Basic I/O Interface

- I/O devices allow the microprocessor to communicate with the outside world.
- Similar to memory interfacing, **I/O interfacing** also uses the data and address lines.
- Data lines are still used to send/receive data from the I/O device.
- However, the address lines are used to specify the I/O address, often called a ***port number***, which specifies the I/O device to be accessed.
- The input/output system of the microprocessor allows peripherals to provide data or receive results of processing the data. This is done using I/O ports.

I/O Instruction

I- There are two basic I/O instructions **IN** and **OUT**.
IN and OUT transfer data between an I/O device and the microprocessor's accumulator (AL, AX) depending on data length.

- **IN** – to read data from I/O device
- **OUT** – to write data to I/O device

- There are two ways to specify the I/O address in IN/OUT instructions:

*1- Fixed address (***Direct I/O instructions)**

- 8-bit address.
- Address specified as an immediate value.
- The address of the I/O port is specified as part of the instruction.
- Intel calls the 8 bit form (p8) a **fixed address** because it is stored with the instruction.

*2- Variable address (***Variable I/O instructions)**

- 16-bit address.
- Address is stored in DX.
- Intel calls the 16 bit form (p16) a **variable address** because it is stored in a DX then used to address the I/O device.

-Whenever data are transferred using IN or OUT instruction, the I/O address (port number) appears on address bus. The external I/O interface decodes it in the same manner that it decodes a memory address.

- The 8-bit **fixed port number** (p8) appears on address bus connections A7-A0 with bits A15-A8 equal to 00000000_2.

- The 16-bit **variable port number** (DX) appears on address bus connections A15-A0.

- The address connections above A_{15} are undefined for an I/O instruction.

- The first 256 I/O port addresses (00-FFh) are accessed by both the fixed and variable I/O instruction.

- But any I/O address from 0100h-FFFFh is accessed only by the variable I/O address.

Ex.
Data are to be read from two byte-wide input ports at addresses $\mathbf{AA_{16}}$ and $\mathbf{A9_{16}}$ and then output as a word to a word-wide output port at address $\mathbf{B000_{16}}$. Write a series of instructions to perform this input/output operation.

```
IN  AL, AAH
MOV AH, AL
IN  AL, A9H
MOV DX, B000H
OUT DX, AX
```

II- There are also instructions **INS** and **OUTS**

- Used to transfer strings of data between memory and I/O device.
- INS and OUTS transfer to I/O devices using ES:DI and DS:SI, respectively.
- Found in all Intel microprocessors except 8086 and 8088.

Address space

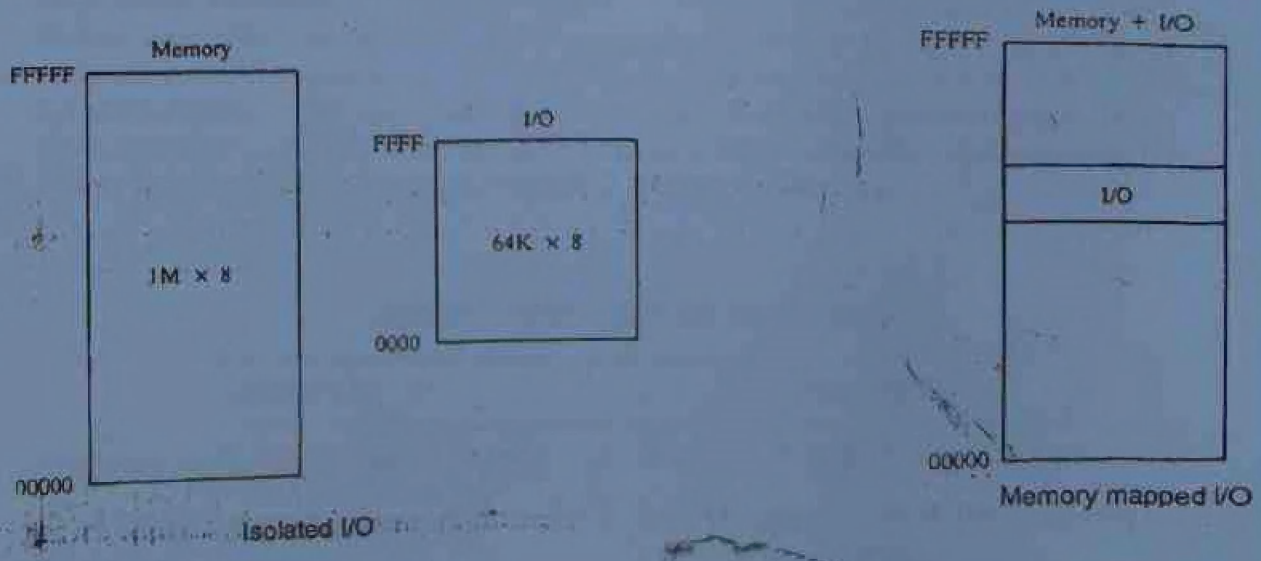
It is defined as the set of all possible addresses that a microprocessor can generate.

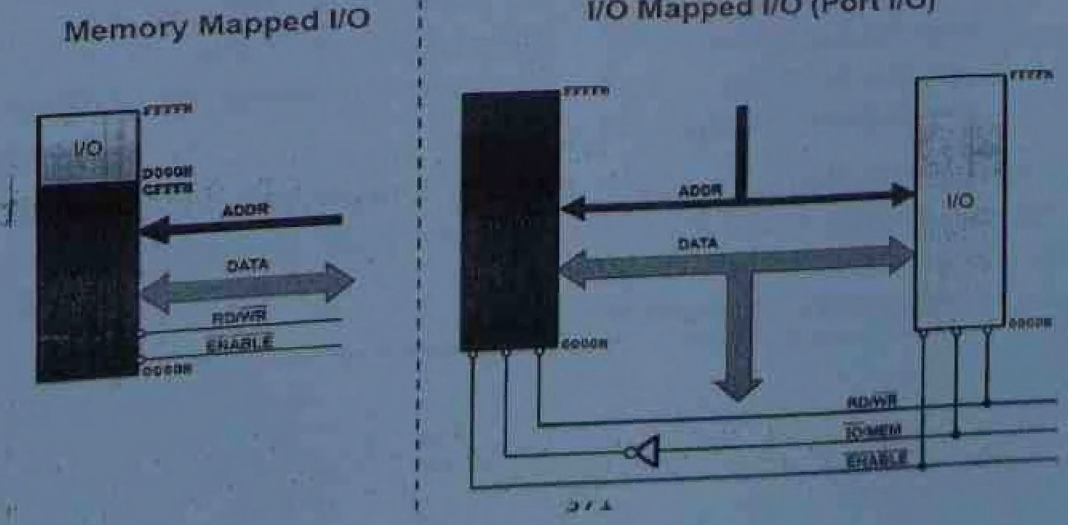
Address space partitioning

8086 microprocessor has a 20-bit address bus so that it can address 2^{20} or 1 MB of address-called the address space of 8086. This total address space can be partitioned or allocated to **memory** or **I/O devices** so that they can be addressed properly. This is called **address space partitioning**.

Types of I/O Interfacing to the microprocessor

The address space can be partitioned in two ways. These are:
1- Memory mapped I/O scheme.
2- I/O mapped I/O scheme (**Isolated** I/O).





Isolated I/O. The most common I/O transfer technique used in the Intel microprocessor-based system is isolated I/O. The term *isolated* describes how the I/O locations are isolated from the memory system in a separate I/O address space. (Figure illustrates both the isolated and memory-mapped address spaces for any Intel 80X86 or Pentium microprocessor.) The addresses for isolated I/O devices, called *ports*, are separate from the memory. As a result, the user can expand the memory to its full size without using any of this space for I/O devices. A disadvantage of isolated I/O is that the data transferred between I/O and the microprocessor must be accessed by the IN, INS, OUT, and OUTS instructions. Separate control signals for the I/O space are developed (using M/$\overline{IO}$ and W/$\overline{R}$) that indicate an I/O read ($\overline{IORC}$) or an I/O write ($\overline{IOWC}$) operation. These signals indicate that an I/O port address appears on the address bus that is used to select the I/O device. In the personal computer, isolated I/O ports are used for controlling peripheral devices. As a rule, an 8-bit port address is used to access devices located on the system board, such as the timer and keyboard interface, while a 16-bit port is used to access serial and parallel ports as well as video and disk drive systems.

Memory-Mapped I/O. Unlike isolated I/O, memory-mapped I/O does not use the IN, INS, OUT, or OUTS instructions. Instead, it uses any instruction that transfers data between the microprocessor and memory. A memory-mapped I/O device is treated as a memory location in the memory map. The main advantage of memory-mapped I/O is that any memory transfer instruction can be used to access the I/O device. The main disadvantage is that a portion of the memory system is used as the I/O map. This reduces the amount of memory available to applications. Another advantage is that the $\overline{IORC}$ and $\overline{IOWC}$ signals have no function in a memory-mapped I/O system and may reduce the amount of circuitry required for decoding.

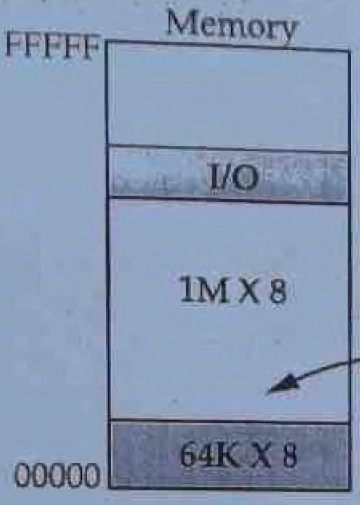
Comparison between isolated and memory mapped I/O

Isolated I/O	Memory mapped I/O
1. I/O devices are treated separate from memory.	1. I/O devices are treated as part of memory.
2. Full 1 MB address space is available for use as memory.	2. Full 1 MB cannot be used as memory since I/O devices are treated as part of memory.
3. Separate instructions are provided in the instruction set to perform isolated I/O input-output operations. These maximise I/O operations.	3. No separate instructions are needed in this case to perform memory mapped I/O operations. Hence, the advantage is that many instructions and addressing modes are available for I/O operations.
4. Data transfer takes place between I/O port and AL or AX register only. This is certainly a disadvantage.	4. No such restriction in this case. Data transfer can take place between I/O port and any internal register. Here, the disadvantage is that it somewhat slows the I/O operations.

Isolated versus Memory-Mapped I/O

Isolated and Memory-Mapped I/O:

- In the Isolated scheme, IN, OUT, INS and OUTS are required.
- In the Memory-mapped scheme, any instruction that references memory can be used.



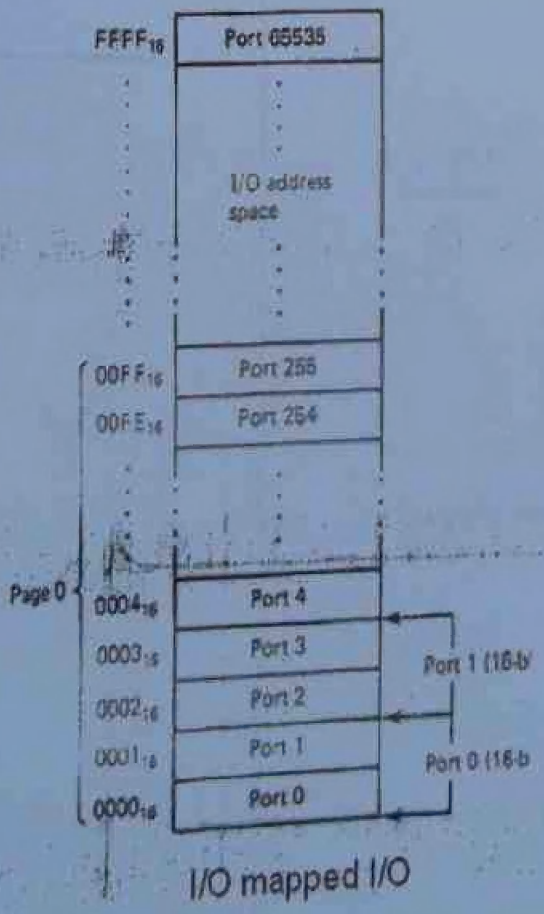
Disadvantage:
A portion of the memory space is used for I/O devices.

Advantage:
$\overline{IORC}$ and $\overline{IOWC}$ not required.
Any data transfer instruction.

Disadvantage:
Hardware using M/$\overline{IO}$ and W/$\overline{R}$ needed to develop signals $\overline{IORC}$ and $\overline{IOWC}$.
Requires IN, OUT, INS and OUTS

8-bit port addresses used to access system board device, e.g. timer and keyboard.

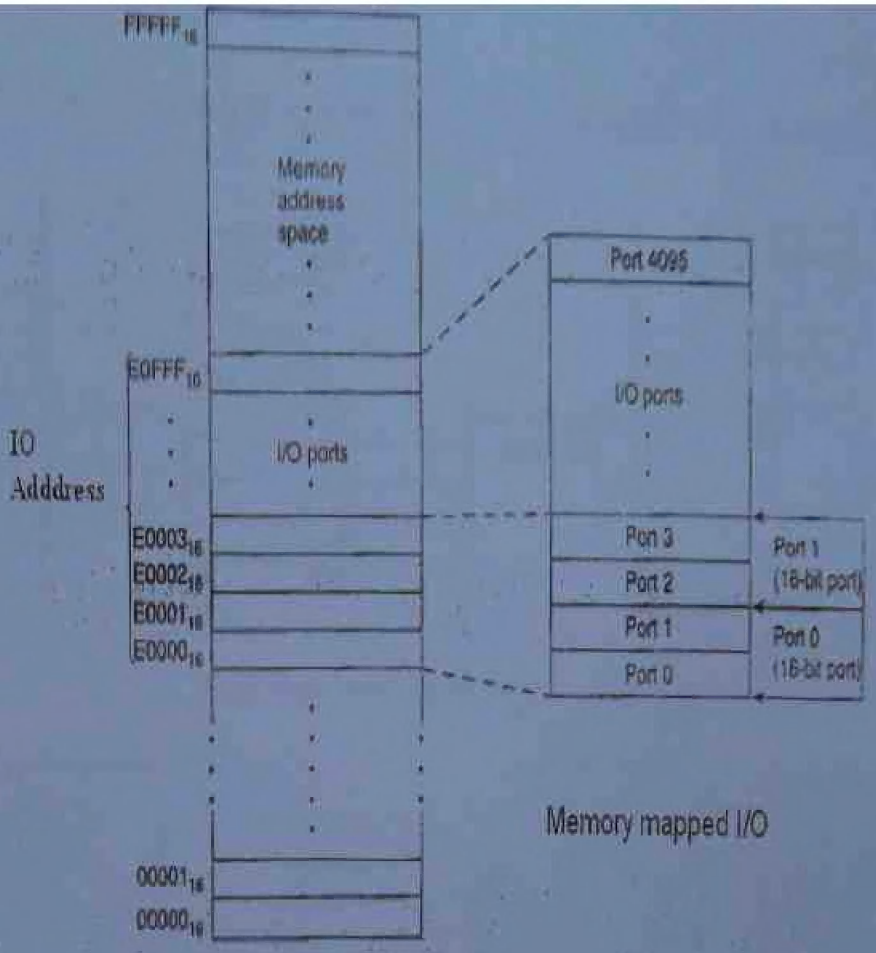
16-bit port addresses used to access serial and parallel ports, harddrives, etc.

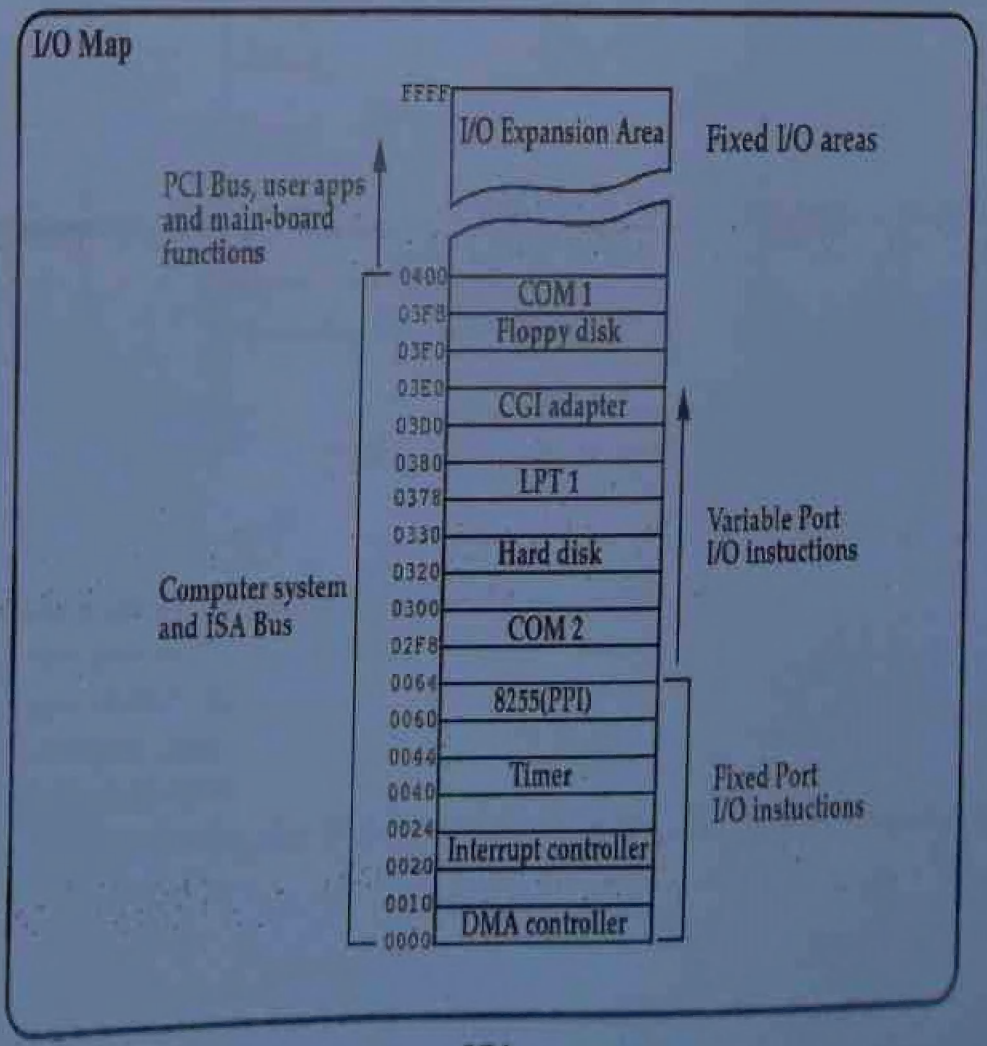


Isolated I/O

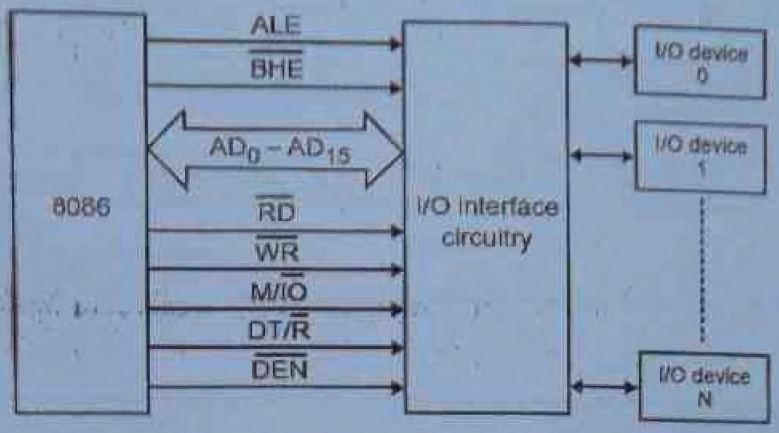

```
MOV     AX, DATA
MOV     DS, AX
MOV     BX, OFFSET
; To Read from a Memory Location
MOV     AL, [BX]

; To Write to a Memory Location
MOV     AL, VALUE
MOV     [BX], AL
```


```
; To Read from PORTA
MOV     DX, PORTA
IN      AL, DX

; To Write to PORTB
MOV     DX, PORTB
MOV     AL, VALUE
OUT     DX, AL
```

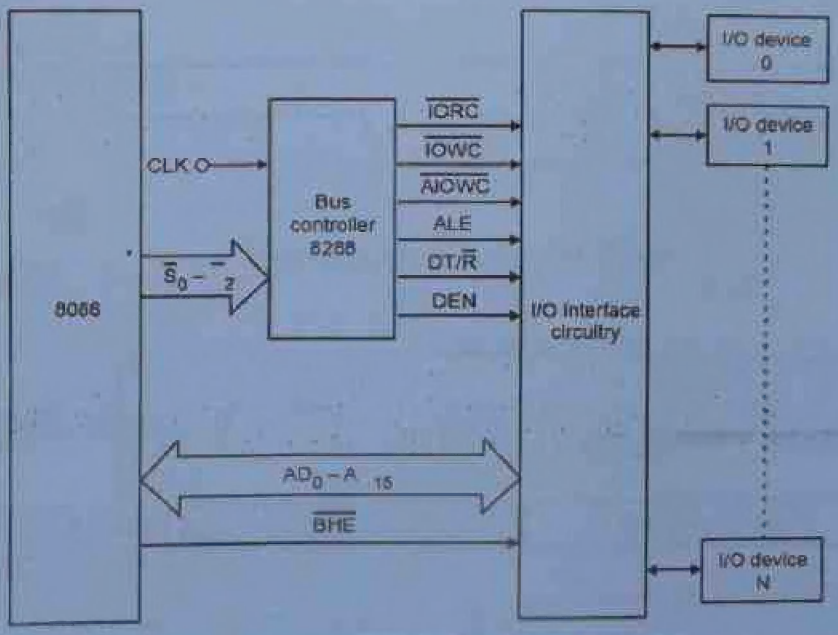

FFFFF16
Memory address space
E0FFF16
I/O ports
IO Adddress
E0003₁₆
E0002₁₆
E0001₁₆
E0000₁₆
00001₁₆
00000₁₆
Port 4095
I/O ports
Port 3
Port 2
Port 1
Port 0
Port 1 (16-bit port)
Port 0 (16-bit port)
Memory mapped I/O


I/O Map
FFFF
I/O Expansion Area
Fixed I/O areas
PCI Bus, user apps and main-board functions
0400
COM 1
03F8
Floppy disk
03F0
03E0
CGI adapter
03D0
0380
LPT 1
0378
0330
Hard disk
0320
Computer system and ISA Bus
0300
COM 2
02F8
Variable Port I/O instuctions
0064
8255(PPI)
0060
0044
Timer
0040
0024
Interrupt controller
0020
0010
DMA controller
0000
Fixed Port I/O instuctions

MIN and MAX mode 8086 based I/O interface



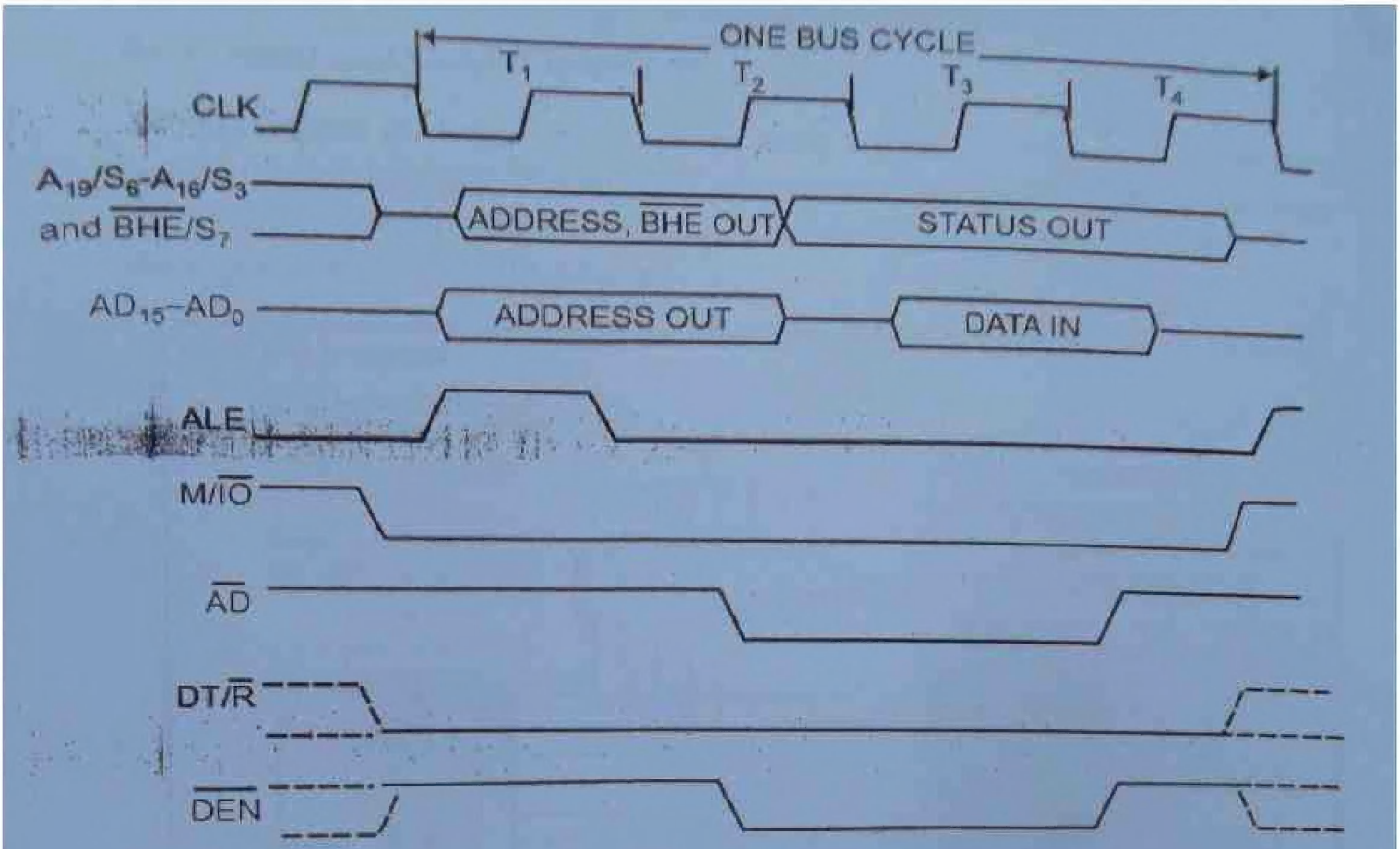
The Minimum mode 8086 system I/O interface



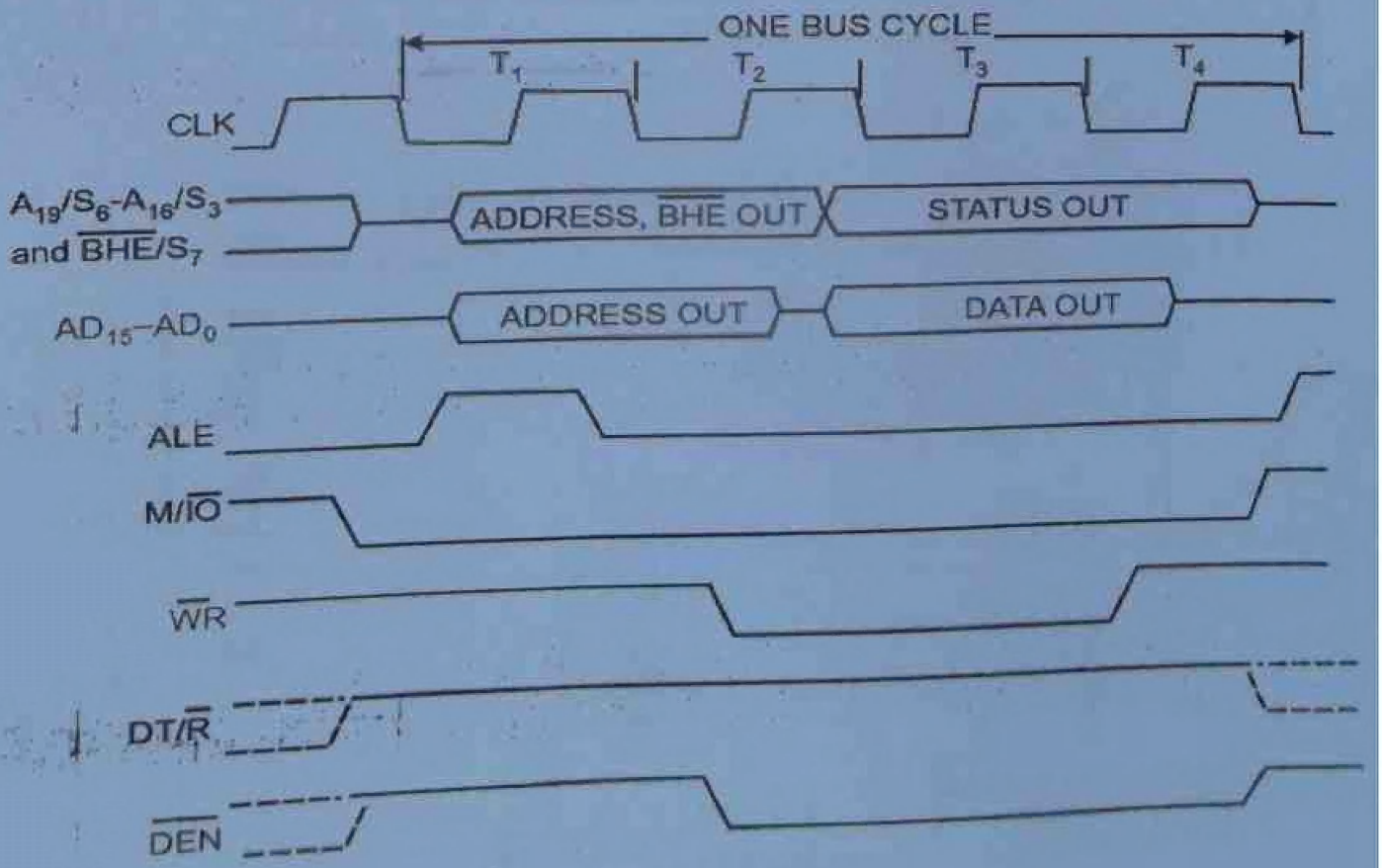
The Maximum-mode 8086 system I/O interface

The interface circuitry performs the following tasks

1. Selecting the particular I/O port
2. Synchronies data transfer
3. Latch the output data
4. Sample the input data
5. Voltage levels between the I/O devices and 8086 are made compatible.



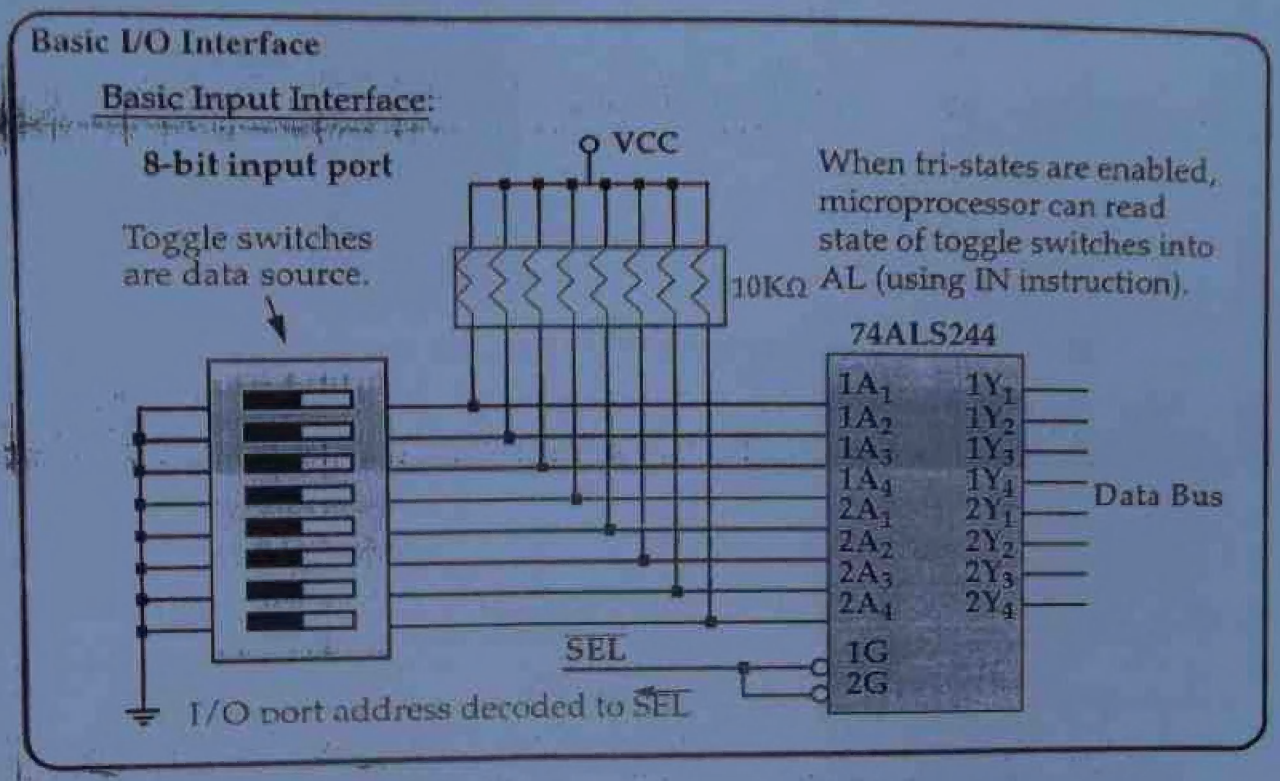
The Input bus cycle of 8086

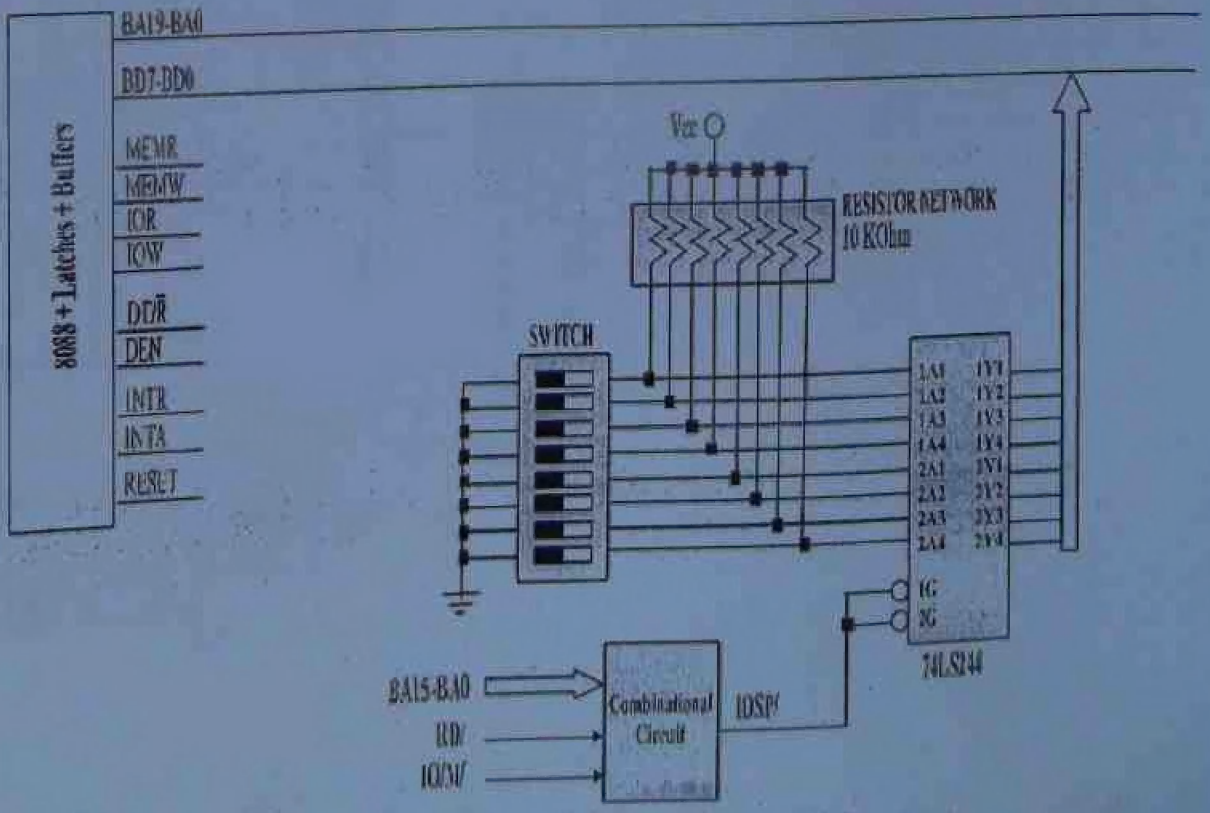


The Output bus cycle of 8086

Basic Input and Output Interfaces

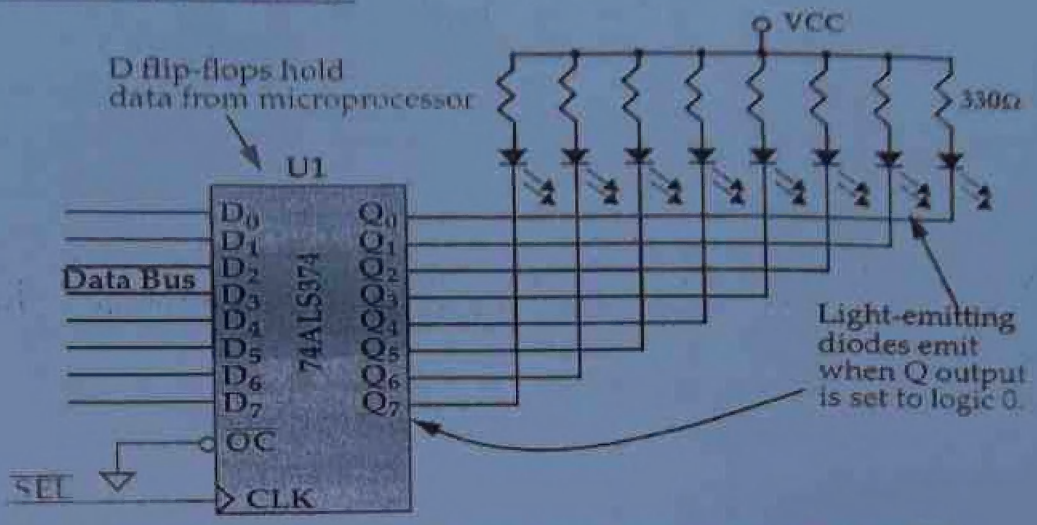
The basic input device is a set of **three-state buffers**. The basic output device is a set of data **latches**. The term **IN** refers to moving data from the I/O device into microprocessor, and the term **OUT** refers to moving data out of the microprocessor to the I/O device.



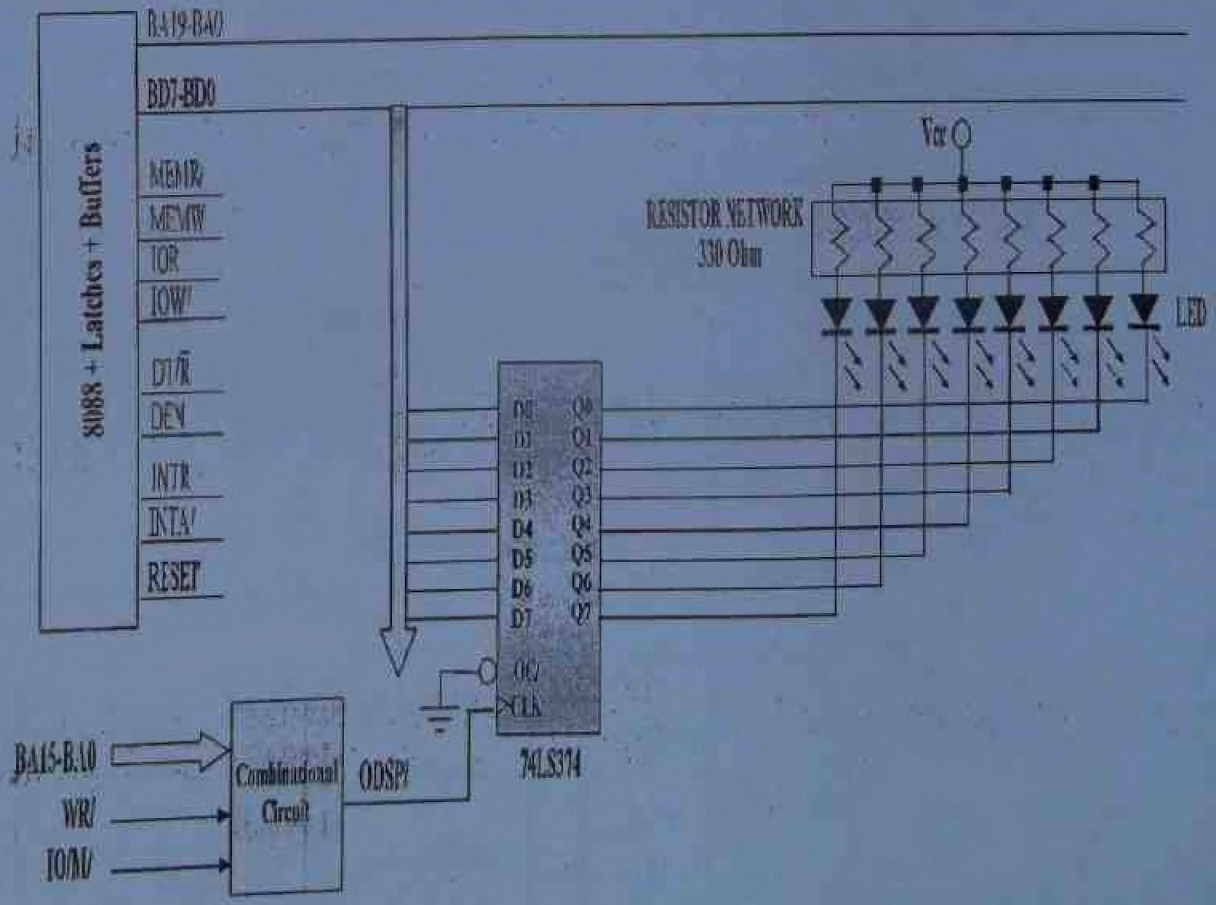


Basic I/O Interface

Basic Output Interface:

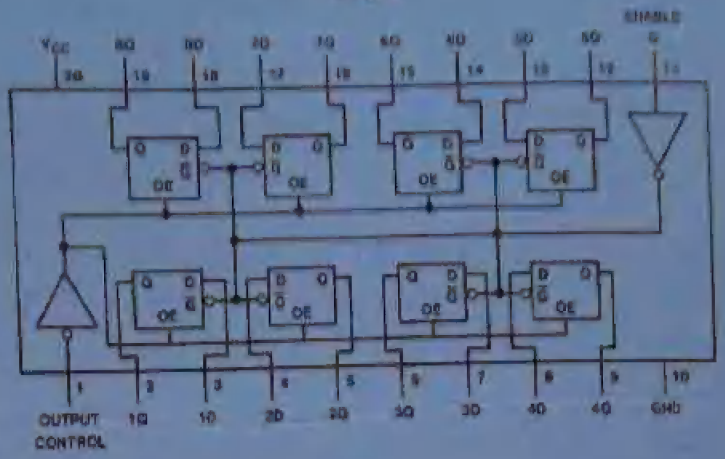


In this case, the data from the OUT instruction is latched using $\overline{SEL}$

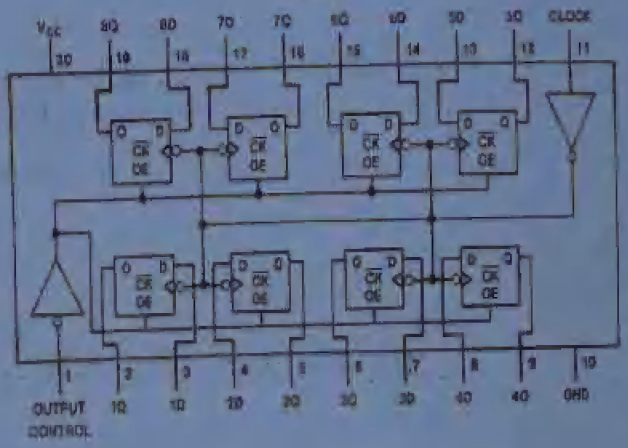


Dual-In-Line Packages

'LS373

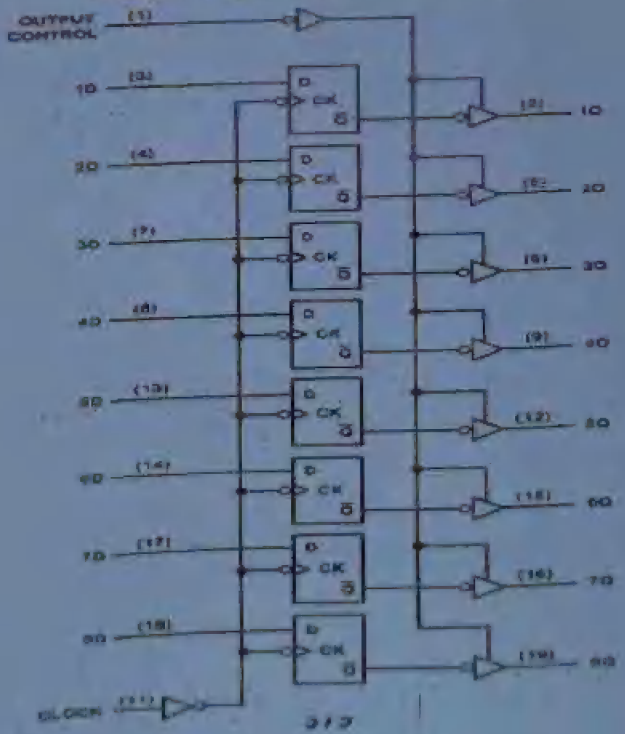


'LS374



DM54/74LS374

Positive-Edge-Triggered Flip-Flops



Time Delay Loop and Blinking an LED at an Output Port

The circuit in figure below show how to attach a LED to output port **Q7** of parallel **port 0**. The port address is 8000H, and the LED corresponds to **bit** 7 of the byte of data that is written to port 0. The circuit use 74LS374 (edge clocked octal latch).

For the LED to turn on, O_7 must b switched to logic 0, and it will remain on until this output is switched back to 1. The 74LS374 is not an inverting latch, therefore, to make O_7 logic 0, simply write 0 to that bit of the octal latch.

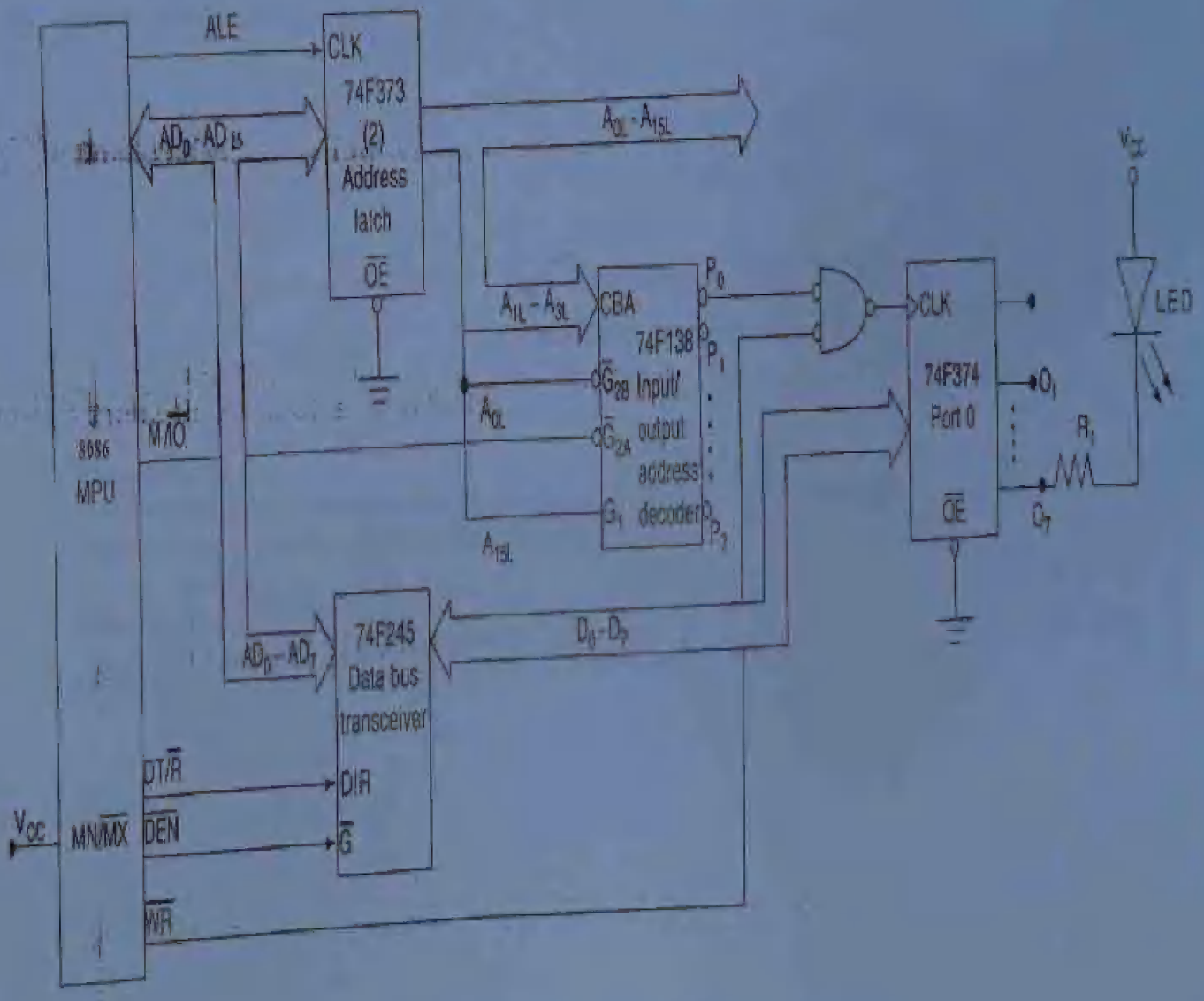


Fig. 1 Driving an LED connected to an output port

Ex :Write instruction sequence to make the LED (in Figure 1.) blink.

Solution: we must write a program that first makes O_7 logic 0 to turn on the LED, delays for a short period of time, and then switches O_7 back to 1 to turn off the LED. This piece of program can run as a loop to make the LED continuously blink. This is done as follows:

```
        MOV DX, 8000H      ; Initialize address of port0
        MOV AL, 00H        ; Load data with bit 7 as logic 0
ON_OFF: OUT DX, AL         ; Output the data to port 0

        MOV CX, FFFFH      ; Load delay count of FFFFH
HERE:   LOOP HERE          ; Time delay loop

        XOR AL, 80H        ; Complement bit 7 of AL
        JMP ON_OFF         ; Repeat to Output the new bit 7
```

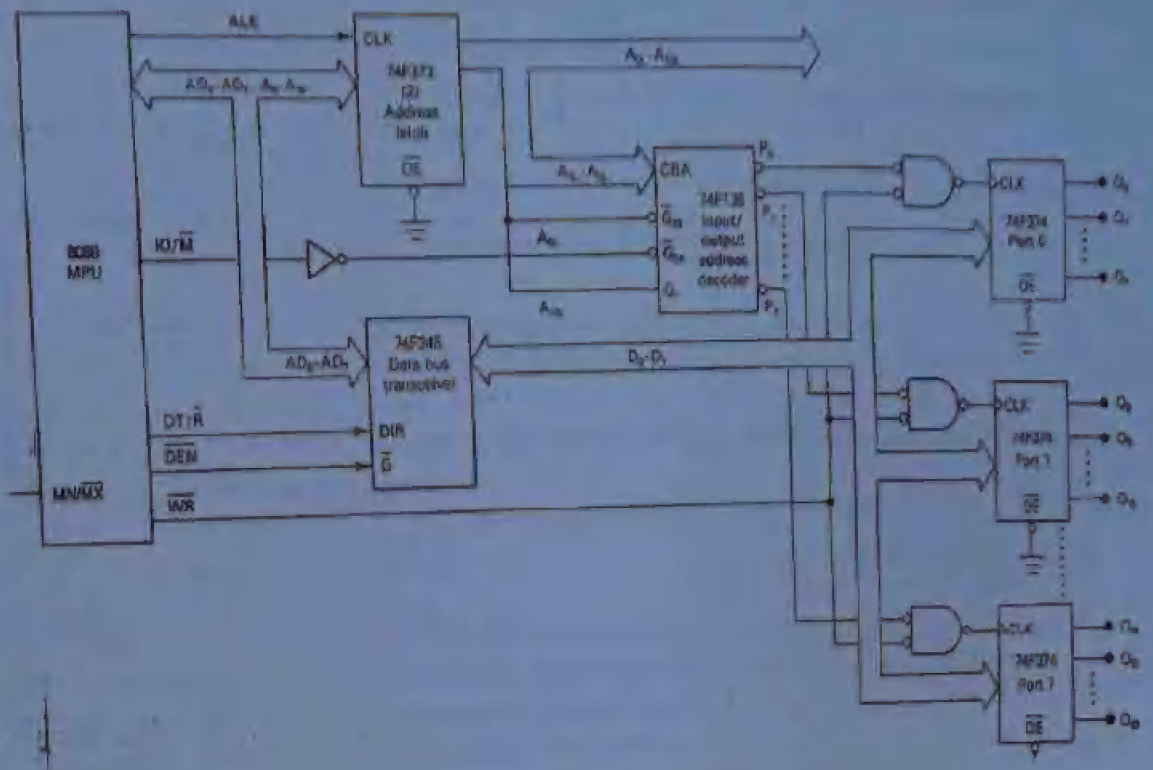
Example :
For the figure 1, what is the I/O address of port 7 on the circuit? Assume all unused address bit are at logic 0.

Solution:

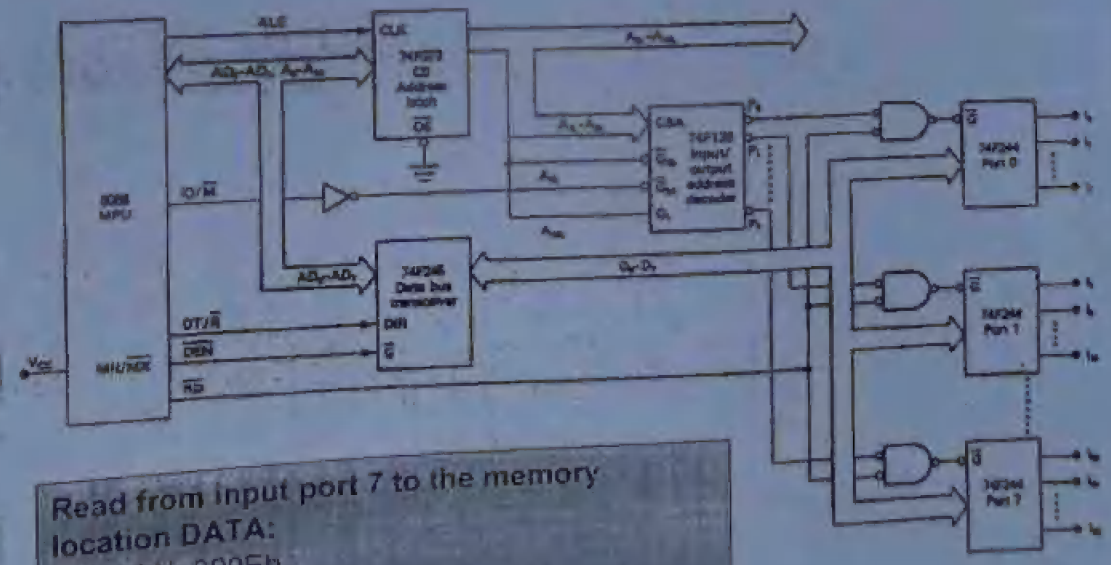
A15	A14	A13	A12	.....	A4	A3	A2	A1	A0
1	0	0	0	.....	0	1	1	1	0

the address is **800E**$_{16}$

Example - 64 line parallel output circuit - 8088



Example - 64 line parallel input circuit



Read from input port 7 to the memory location DATA:

```
MOV DX, 800Eh
IN AL, DX
MOV DATA, AL
```

Handshaking

Many I/O devices accept or release information at a much slower rate than the microprocessor. Another method of I/O control, called *handshaking* or *polling*, synchronizes the I/O device with the microprocessor. An example device that requires handshaking is a parallel printer that prints 100 characters per second (CPS). It is obvious that the microprocessor can definitely send more than 100 CPS to the printer, so a way to slow the microprocessor down to match speeds with the printer must be developed.

Handshaking

I/O devices are typically slower than the microprocessor.

Handshaking is used to synchronize I/O with the microprocessor

A device indicates that it is ready for a command or data (through some I/O pin or port).

The processor issues a command to the device, and the device indicates it is busy (not ready).

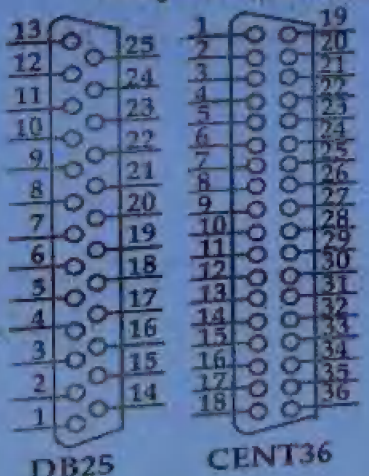
The I/O device finishes its task and indicates a ready condition, and the cycle continues.

There are two basic mechanisms for the processor to service a device.

- Polling: Processor initiated. Device indicates it is ready by setting some status bit and the processor periodically checks it.
- Interrupts: Device initiated. The act of setting a status bit causes an interrupt, and the processor calls an ISR to service the device.

A printer connected to the parallel port requires handshaking.

The parallel port specification is shown below:



DB25	CENT36	Function	DB25	CENT36	Function
1	1	Data Strobe	12	12	Paper out
2	2	Data0	13	13	Select
3	3	Data1	14	14	Afd
4	4	Data2	15	32	Error
5	5	Data3	16	-	RESET
6	6	Data4	17	31	Select in
7	7	Data5	18-25	19-30	GND
8	8	Data6	-	17	Frame GND
9	9	Data7	-	16	GND
10	10	Ack	-	33	GND
11	11	Busy			

The processor writes ASCII data out to the Datax pins of the printer and toggles the Data Strobe pin to latch it in.

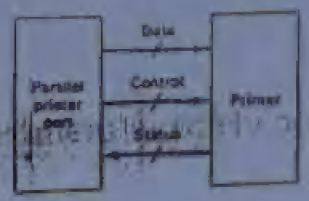
The printer raises the Busy pin.

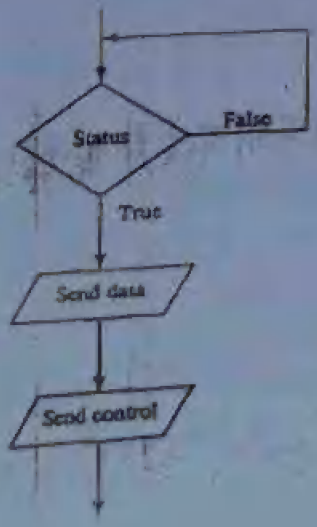
When the Busy pin goes low, the sequence repeats.

Input Output Handshaking

- The I/O ports of a computer typically operate at different data rates
- A hard disk drive, for example, might require the computer to input data at 10Mbps → 100Mbps
- CD-ROM drives operate at 300-600 Kbps
- However when inputting keystrokes from the operator, the data rate may fall to only one or two characters per sec.
- If the processor is to operate efficiently, one needs to develop a strategy to control or synchronize the flow of data between the processor and the widely varying rates of its I/O devices
- This type of synchronization is achieved by implementing what is known as handshaking as part of the input/output interface
- Printers typically have buffers that can be filled by the computer at high speed
- Once full the computer must wait while the data in the buffer is printed
- Most printer manufacturers have settled on a standard set of data and control signals Centronics Parallel Printer Interface

Parallel Printer Interface





Pin	Assignment
1	Strobe
2	Data 0
3	Data 1
4	Data 2
5	Data 3
6	Data 4
7	Data 5
8	Data 6
9	Data 7
10	Ack
11	Busy
12	Paper Empty
13	Select
14	Auto Foxt
15	Error
16	Initialize
17	Slctin
18	Ground
19	Ground
20	Ground
21	Ground
22	Ground
23	Ground
24	Ground
25	Ground

ACK is used by printer to acknowledge receipt of data and can accept a new character.

BUSY high if printer is not ready to accept a new character

SELECT when printer is turned on

ERROR goes low when there are conditions such as paper jam, out of paper, offline

STROBE when PC presents a character

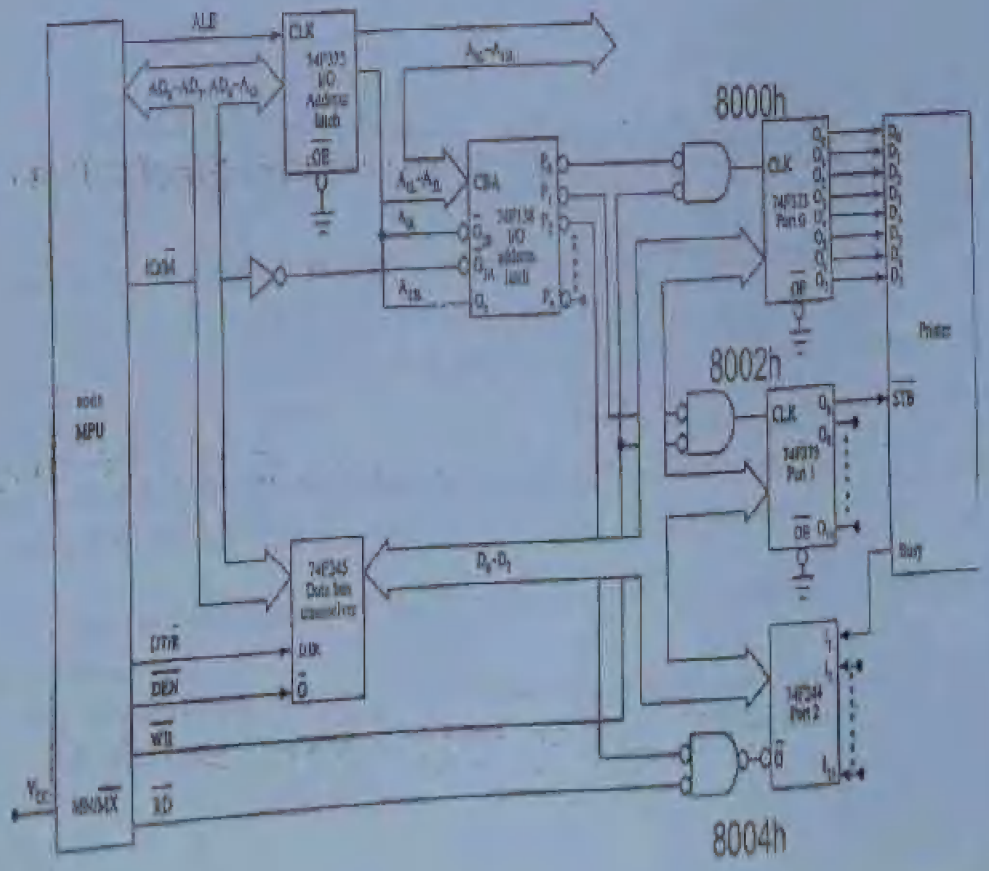
INITIALIZE Clear Printer Buffer and reset control

Operational Principle - Parallel Printer Port

- The computer checks the BUSY signal from the printer, if not BUSY then
- When the PC presents a character to the data pins of the printer, it activates the STROBE pin, telling it that there is a byte sitting at the data pins. Prior to asserting STROBE pin, the data must be at at the printer's data pins for at least 0.5 microsec. (data setup time)
- The STROBE must stay for 0.5 microsec

- The printer asserts BUSY pin indicating the computer to wait
- When the printer picks up the data, it sends back the ACK signal, keeps ACK low for 5 microsec.
- As the ACK signal is going high, the printer makes the BUSY pin low to indicate that it is ready to accept the next byte
- The CPU can use ACK or BUSY signals from the printer to initiate the process of sending another byte

Printer Interface Circuit



I/O port Address Decoding

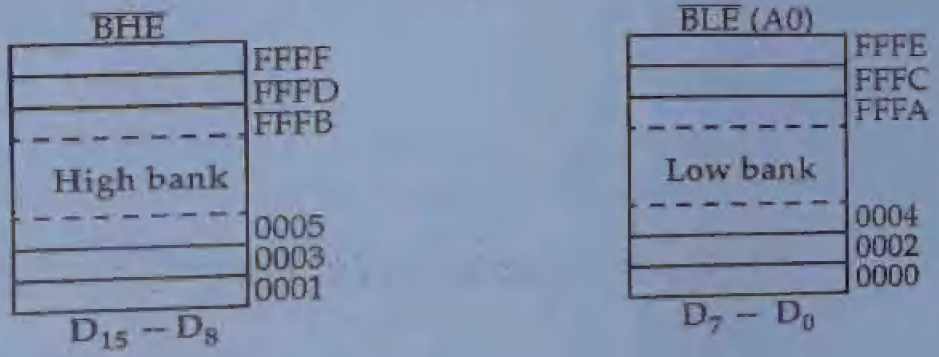
- I/O port address decoding is very similar to memory address decoding, especially for memory mapped I/O devices.

- The main difference between memory decoding and isolated I/O decoding is the number off address pins connected to the decoder. We decode A19-A0 for memory decoding and A15-A0 for isolated I/O.- If the I/O devices use only fixed I/O addressing devices, we decode only A7-A0.

- Another difference is that we use the IORC and IOWC to activate I/O devices for a read or a write operation.

For isolated I/O, $\overline{IORC}$ and $\overline{IOWC}$ are developed using M/$\overline{IO}$ and W/$\overline{R}$ pins of the microprocessor.

The I/O banks on the 8086 through the 80386SX are also set up like the memory.



Output devices can be 16-bit in which case $\overline{BHE}$ is not needed.

Input devices can be 8-bit or 16-bit.

Note that instead of latches, high impedance buffers (74ALS244) are used in these cases.

32-bit ports are becoming more popular because of PCI bus primarily.

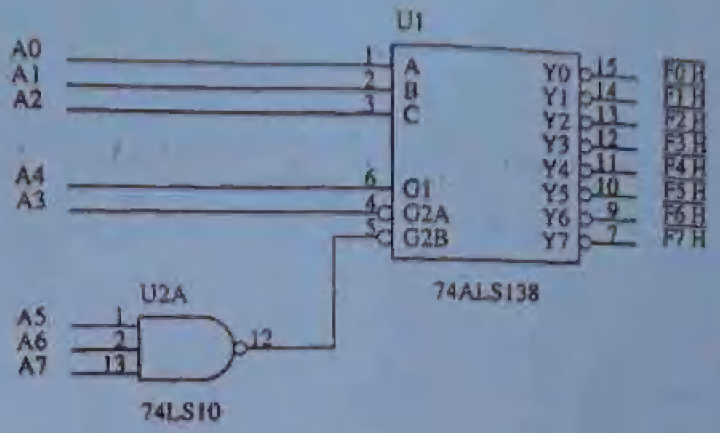
The EISA and VESA local bus are also 32-bit buses.

For the 64-bit data buses of the Pentium, the I/O ports can appear in any of the 8 banks.

However, only 32-bit transfers are supported, as there are no 64-bit transfer instructions.

Decoding 8-Bit I/O Addresses

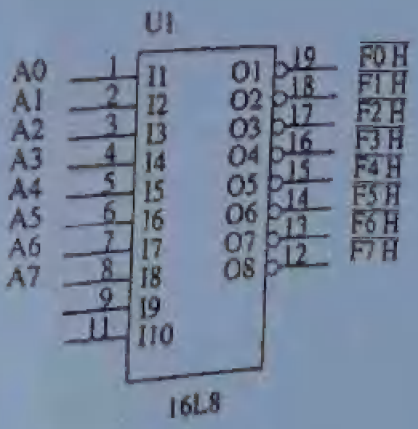
As mentioned, the fixed I/O instruction uses an 8-bit I/O port address that appears on A_{15}–A_0 as 0000H–00FFH. If a system will never contain more than 256 I/O devices, we often decode only address connections A_7–A_0 for an 8-bit I/O port address. Thus, we ignore address connections A_{15}–A_8.



A port decoder that decodes 8-bit I/O ports. This decoder generates active low outputs for ports F0H–F7H

The figure below shows the PAL version of this decoder. Notice that this is a better decoder circuit because the number of integrated circuits has been reduced to one device.

```
CHIP      DECODER8 PAL16L8

;pins  1  2  3  4  5  6  7  8  9  10
       A0 A1 A2 A3 A4 A5 A6 A7 NC GND

;pins 11 12 13 14 15 16 17 18 19  20
       NC F7 F6 F5 F4 F3 F2 F1 F0 VCC

EQUATIONS

/F0 = A7 * A6 * A5 * A4 * A3 * /A2 * /A1 * /A0
/F1 = A7 * A6 * A5 * A4 * A3 * /A2 * /A1 * A0
/F2 = A7 * A6 * A5 * A4 * A3 * /A2 * A1 * /A0
/F3 = A7 * A6 * A5 * A4 * A3 * /A2 * A1 * A0
/F4 = A7 * A6 * A5 * A4 * A3 * A2 * /A1 * /A0
/F5 = A7 * A6 * A5 * A4 * A3 * A2 * /A1 * A0
/F6 = A7 * A6 * A5 * A4 * A3 * A2 * A1 * /A0
/F7 = A7 * A6 * A5 * A4 * A3 * A2 * A1 * A0
```

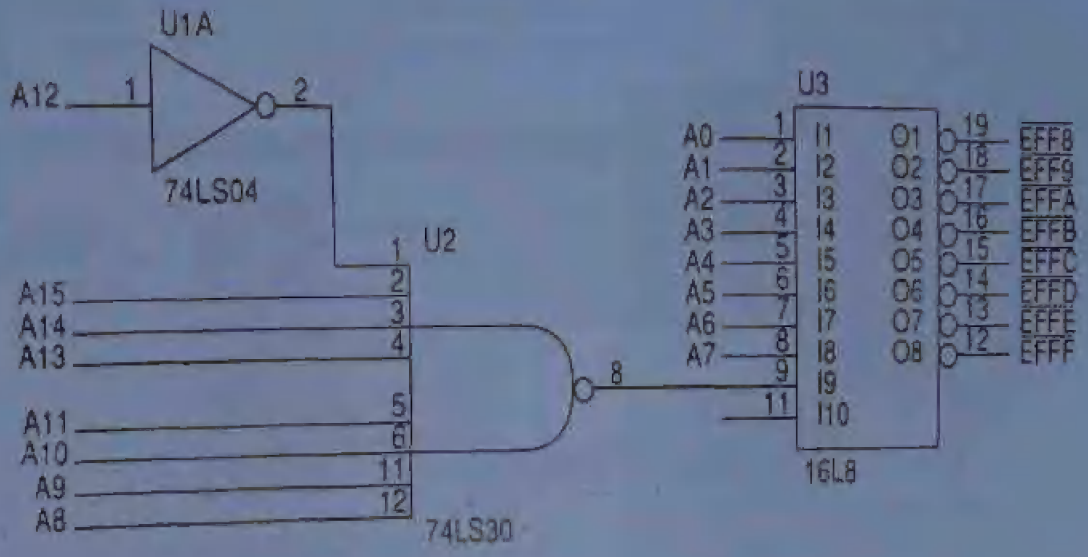


PAL16L8 decoder that generates I/O port signals for port F0H–F7H

Decoding 16-bit I/O Address

- The main difference between decoding an 8-bit I/O address and a 16-bit I/O address is that eight additional address lines (A15-A8) must be decoded.

- Figure below illustrates a circuit that contains a PAL16Lb and an 8-bit NAND gate used to decode I/O ports EFF8H-EFFFH. The NAND gate decodes (A15-A8) to generate a signal to enable the PAL16L8 for any I/O address between EF00 and EFFF.

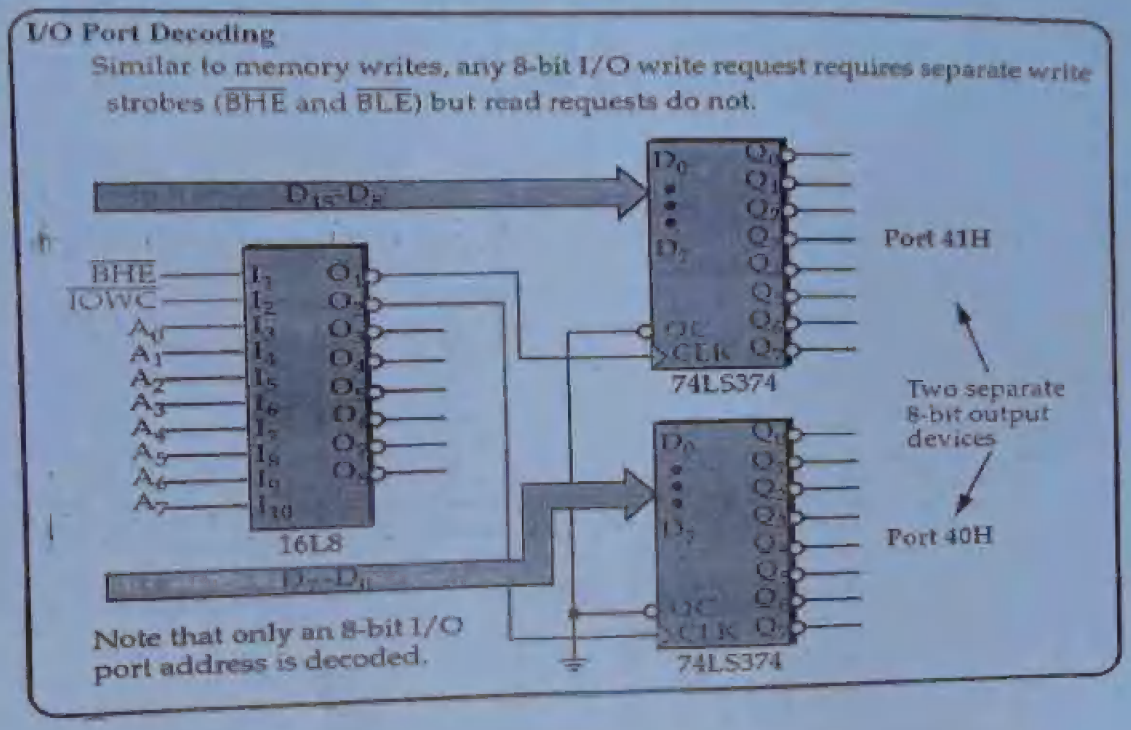



```
CHIP      DECODER9 PAL16L8

:pins   1  2  3  4  5  6  7  8  9     10
        A0 A1 A2 A3 A4 A5 A6 A7 NAND GND

:pins 11     12     13     14     15     16     17     18     19  20
      NC EFFFH EFFEH EFFDH EFFCH EFFBH EFFAH EFF9H EFF8H VCC

EQUATIONS

/EFF8H = A7 * A6 * A5 * A4 * A3 * /A2 * /A1 * /A0 * /NAND
/EFF9H = A7 * A6 * A5 * A4 * A3 * /A2 * /A1 * A0 * /NAND
/EFFAH = A7 * A6 * A5 * A4 * A3 * /A2 * A1 * /A0 * /NAND
/EFFBH = A7 * A6 * A5 * A4 * A3 * /A2 * A1 * A0 * /NAND
/EFFCH = A7 * A6 * A5 * A4 * A3 * A2 * /A1 * /A0 * /NAND
/EFFDH = A7 * A6 * A5 * A4 * A3 * A2 * /A1 * A0 * /NAND
/EFFEH = A7 * A6 * A5 * A4 * A3 * A2 * A1 * /A0 * /NAND
/EFFF7 = A7 * A6 * A5 * A4 * A3 * A2 * A1 * A0 * /NAND
```

8- and 16-Bit I/O Ports

Now that we understand that decoding the I/O port address is probably simpler than decoding a memory address (because of the number of bits), we explain how data are transferred between the microprocessor and 8- or 16-bit I/O devices. Data transferred to an 8-bit I/O device exist in one of the I/O banks in a 16-bit microprocessor such as the 8086, 80286, 80186, or 80386SX.

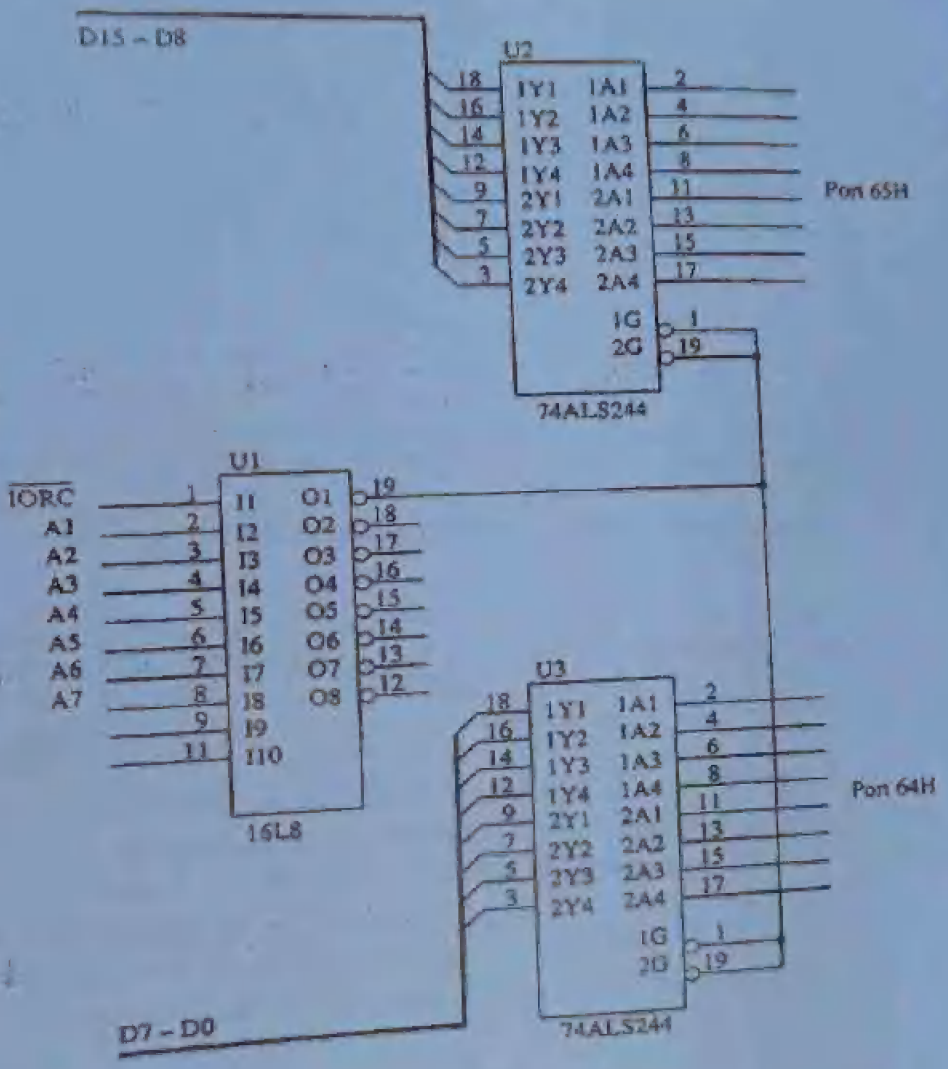
Figure Bellow illustrates a system that contains two different 8-bit output devices located at 8-bit I/O address 40H and 41H. Because these are 8-bit devices and because they appear in different I/O banks, we generate separate I/O write signals.




```
;pins  1     2  3  4  5  6  7  8  9  10
       BHE IOWC A0 A1 A2 A3 A4 A5 A6 GND

;pins 11 12 13 14 15 16 17 18 19 20
       A7 NC NC NC NC NC NC 40 41 VCC

EQUATIONS

/40 = /BLE * /IOWC * /A7 * A6 * /A5 * /A4 * /A3 * /A2 * /A1
/41 = /BHE * /IOWC * /A7 * A6 * /A5 * /A4 * /A3 * /A2 * /A1
```

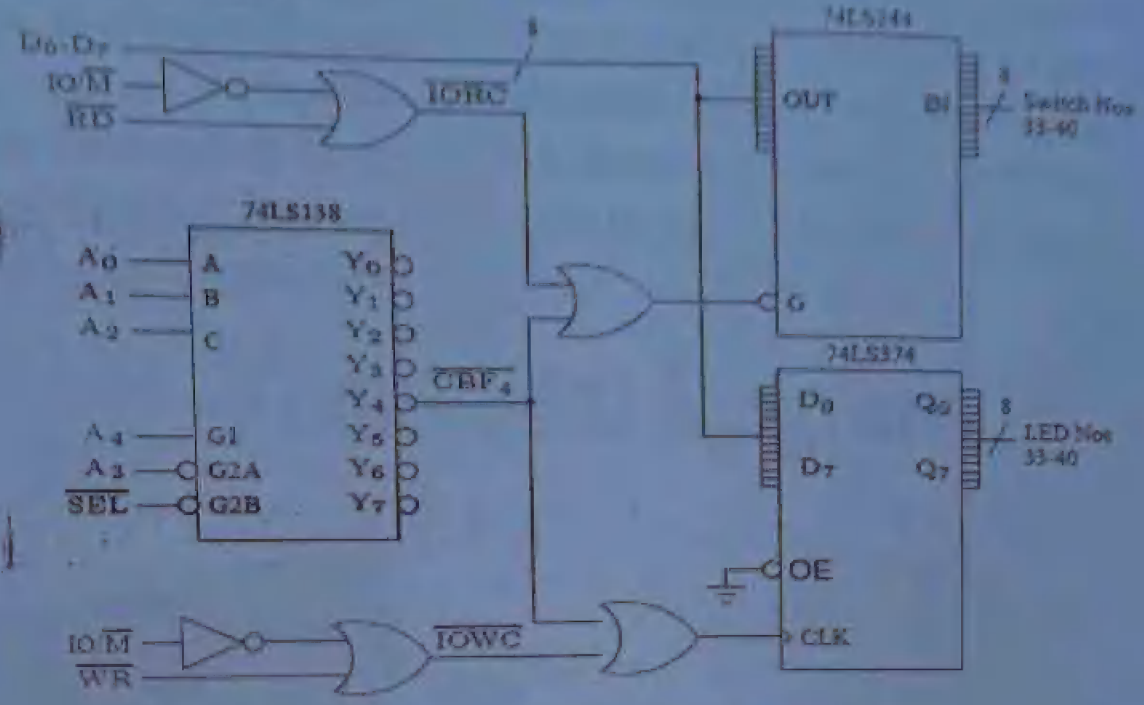
When selecting 16-bit wide I/O devices, the $\overline{BLE}$ (A0) and $\overline{BHE}$ pins have no function because both I/O banks are selected together. Although 16-bit I/O devices are relatively rare, a few do exist for analog-to-digital and digital-to-analog converters, as well as for some video and disk memory interfaces.

Figure Bellow illustrates a 16-bit input device connected to function at 8-bit I/O addresses 64H and 65H. Notice that the PAL16L8 decoder does not have a connection for address bit $\overline{BLE}$ (A0) and $\overline{BHE}$ because these signals do not apply to 16-bit wide I/O devices.




```
CHIP     DECODERB PAL16L8

;pins   1  2  3  4  5  6  7  8  9  10
       IORC A1 A2 A3 A4 A5 A6 A7 NC GND

;pins 11 12 13 14 15 16 17 18 19  20
      NC NC NC NC NC NC NC NC O6X VCC

EQUATIONS

/O6X = /IORC * /A7 * A6 * A5 * /A4 * /A3 * /A2 * /A1
```

Interface Design



Only one input and one output bank at address CBF4H is shown

```
Operation
MOV DX, CBF4h    ; Load port address.
IN AL, DX        ; Read switches.
OUT DX, AL       ; Update LED's
```

Read:

(a) T1: CBF4h → Address Lines, '1' → IO/M.
(b) '138 enabled with A2A1A0 = 100; Y4 ← '0'
(c) 74LS244 enabled → switch data appears on D7-D0
(d) T2: '0' → RD
(e) T3: Data read by µP at end of T3.
(f) T4: '1' → RD, address & control de-asserted by µP

```
MOV DX, CBF4H    ; Load port address.
IN AL, DX        ; Read switches.
OUT DX, AL       ; Update LED's
```

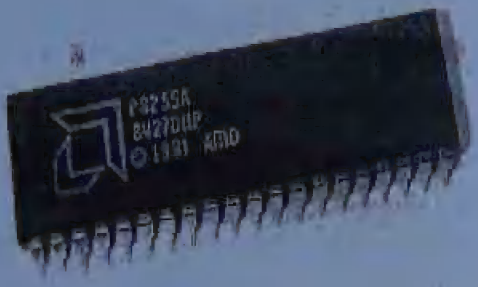
Write:

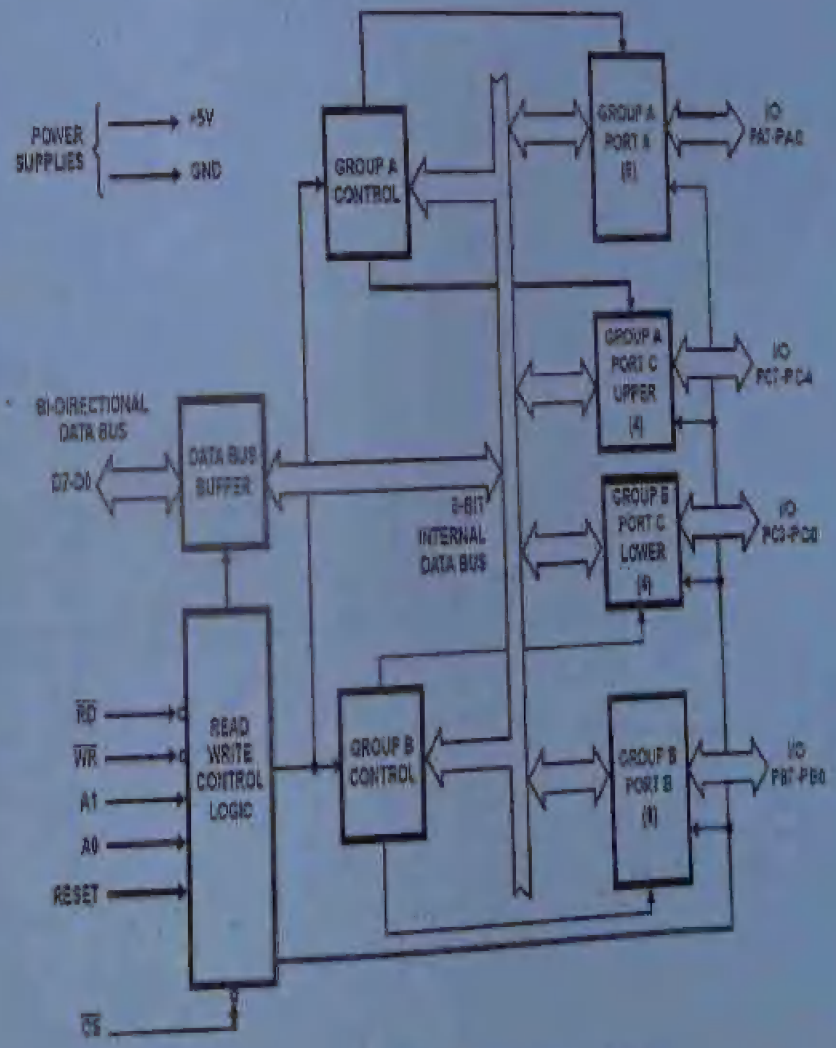
(a) T1: CBF4h → Address Lines, '1' → IO/M
(b) '138 enabled with A2A1A0 = 100; Y4 ← '0'
(c) T2: '0' → WR, Data on D7-D0
(d) T4: '1' → WR, clocks '374. D-inputs to Q-outputs.
Address, data & control de-asserted by µP.

THE PROGRAMMABLE PERIPHERAL INTERFACE

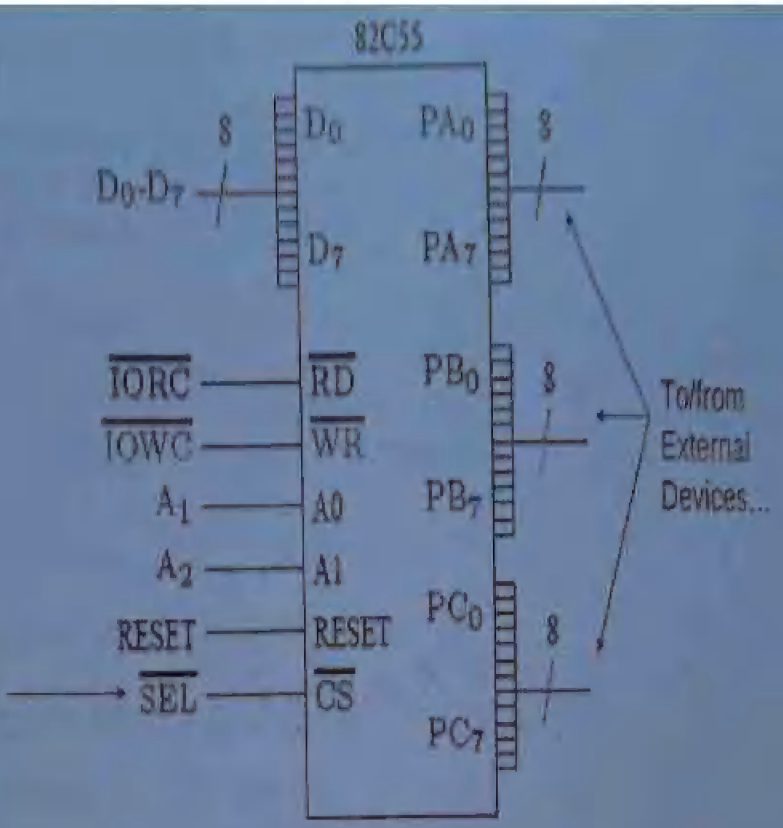
The 82C55 **programmable peripheral interface** (PPI) is a very popular low-cost interfacing component found in many applications. The PPI has 24 pins for I/O, programmable in groups of 12 pins, that are used in three separate modes of operation. The 82C55A can interface any TTL-compatible I/O device to the microprocessor. The 82C55 is used for interface to the keyboard and the parallel printer port in many of these personal computers. It also controls the timer and reads data from the keyboard interface.

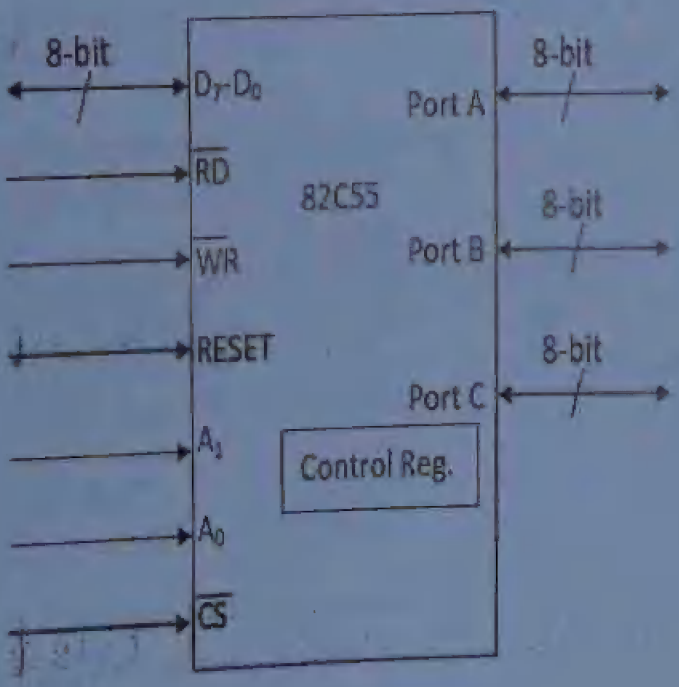
Functional Diagram

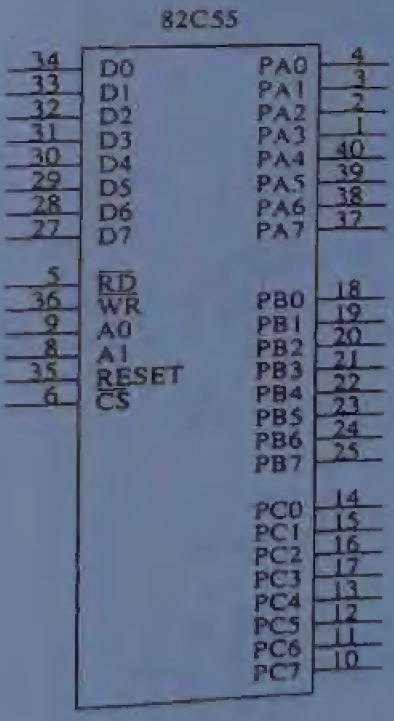




Block diagram of 8086







Pin diagram of PPI 8255

82C55A Programmable Peripheral Interface

- The 82C55A is LSI peripheral designed to permit easy implementation of parallel I/O in the 8086microcomputer systems.

- It provides a flexible parallel interface, which includes features such as single-bit, 4-bit, and byte-wide input and output ports; Level-sensitive inputs; latched outputs; strobed inputs or outputs; and strobed bidirectional input/output these features are selected under software control.

- It has three ports, each contains eight lines.

- **Port A:** It can be programmed in 3 modes – mode 0, mode 1, mode 2.
- **Port B:** It can be programmed in mode 0, mode1.
- **Port C:** This port can be divided into two 4 bit ports and can be used as control signals for port A and port B. It can be programmed in mode 0.

- It has two groups of I/O pins were named as **Group A** and **Group B**.

- Group A contains an 8-bit port A (PA0- PA7) along with a 4-bit port C upper (PC4 - PC7).

- Group B contains an 8-bit port B (PB0 –PB7) along with a 4-bit port C Lower (PC0 - PC3).

- These ports can function independently either as input or as output ports using CWR (Control Word Register).

- There are six pins associated with Read/Write control logic block. These are RD, WR, A0, A1, Reset and CS signals.

PA0-PA7: These are eight port A lines that acts as either latched output or buffered input lines depending upon the control word loaded into the CWR.

- **PC4-PC7** : Upper nibble of port C lines. They may act as either output latches or input buffers lines. This port also can be used for generation of handshake lines in mode 1 or mode 2.

- **PC0-PC3** : These are the lower port C lines, other details are the same as PC7-PC4 lines.

• **PB0-PB7** : These are the eight port B lines which are used as latched output lines or buffered input lines .

• **RD'**: This is the input line driven by the microprocessor, indicating read operation.

• **WR'**: This is an input line driven by the microprocessor, indicating write operation.

D0-D7 : These are the data bus lines those carry data or control word to/from the microprocessor.

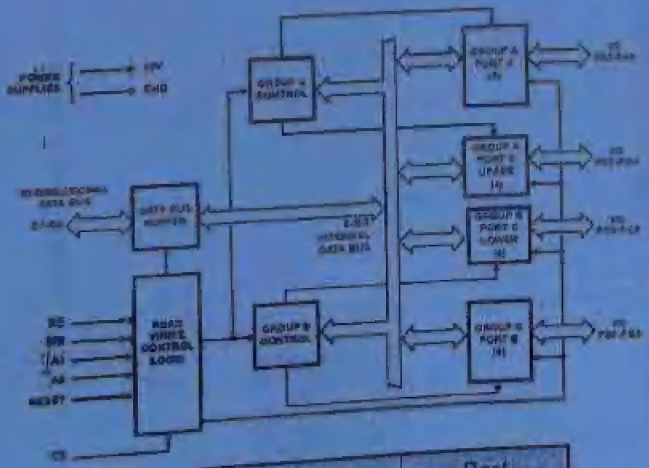
• **RESET** : A logic high on this line clears the control word register of 8255. All ports are set as input ports by default after reset.

• **CS'** : This is a chip select line

• **A0-A1** : These are the address input lines and are driven by the microprocessor.

These address lines are used for addressing any one of the four registers.

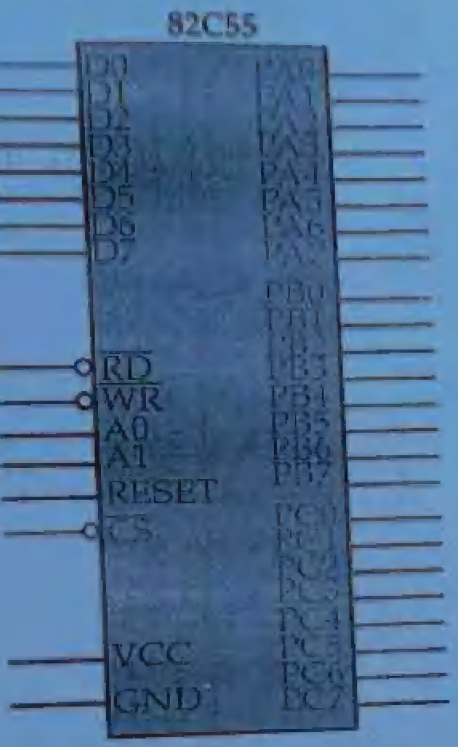
8255A Basic Operation



A_1	A_0	Port
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control Word

A1	A0	RD	WR	CS	INPUT OPERATION (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					DISABLE FUNCTION
X	X	X	X	1	Data Bus → Three-State
X	X	1	1	0	Data Bus → Three-State

Pinout of 82C55 PPI



Group A

Port A (PA7-PA0) and upper half of port C (PC7 - PC4)

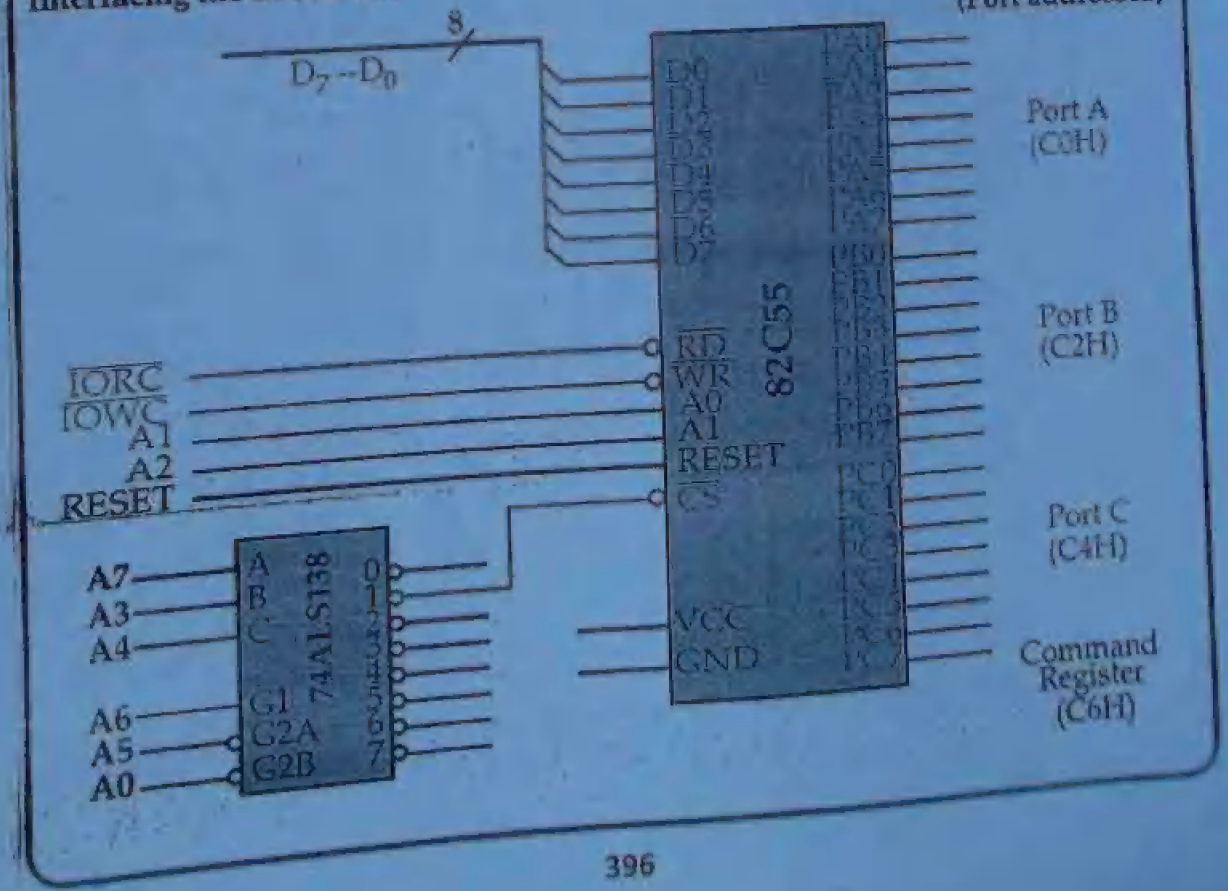
Group B

Port B (PB7-PB0) and lower half of port C (PC3 - PC0)

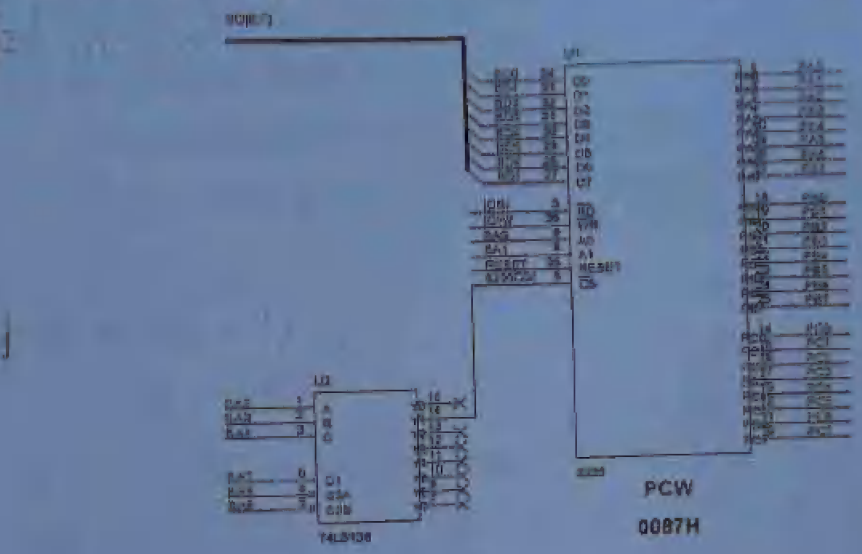
I/O Port Assignments

A_1	A_0	Function
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Command Register

Interfacing the 82C55 PPI

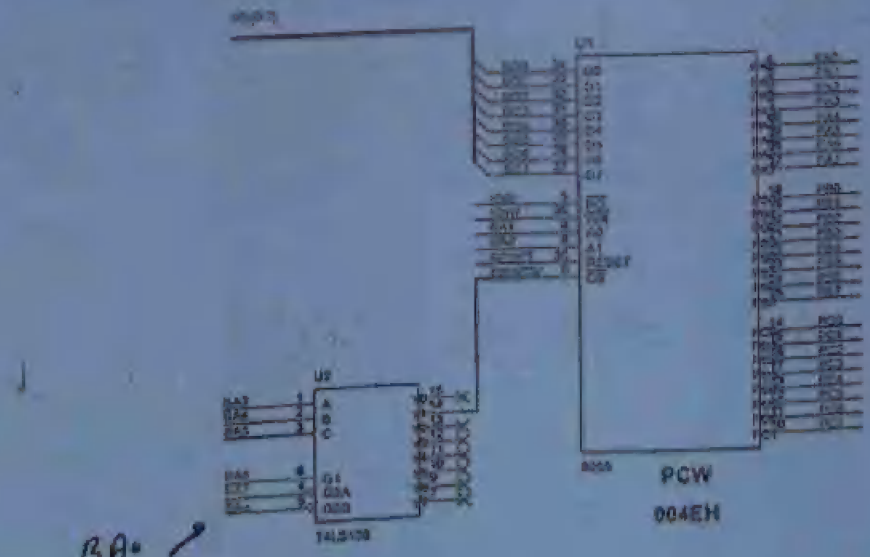


Interfacing 8255A to Buffered 8088 System



BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
x	x	x	x	x	x	x	x	1	0	0	0	0	1		
								Enable Signals			Y1			Ports	

Interfacing 8255A to Buffered 8086 System



BA0

BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
x	x	x	x	x	x	x	x	0	1	0	0	1			0
								Enable Signals			Y1			Ports	Enable

Decoding Circuit for I/O Chips

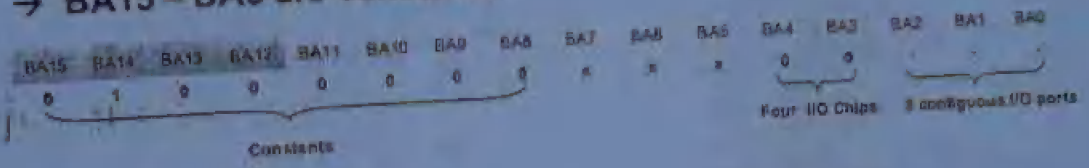
Example :

- A system has 4 I/O chips.
- The dedicated I/O space is: 4000H – 40FFH.
- Some of the I/O chips require 8 contiguous I/O ports.

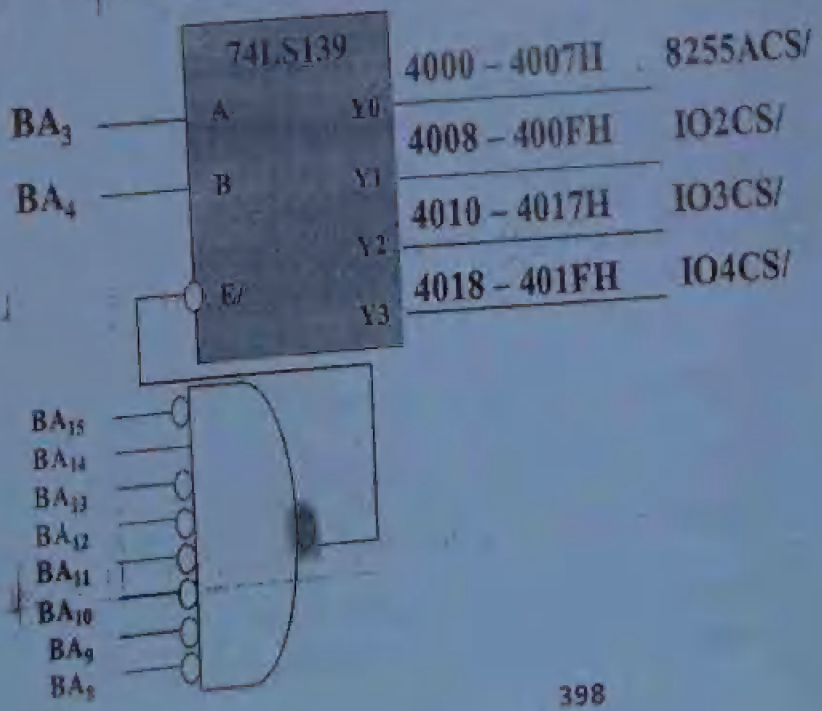
Solution:

4000 – 40FF

→ BA15 – BA8 are constants.



Solution of Example :



Port A	4000H
Port B	4001H
Port C	4002H
CW	4003H

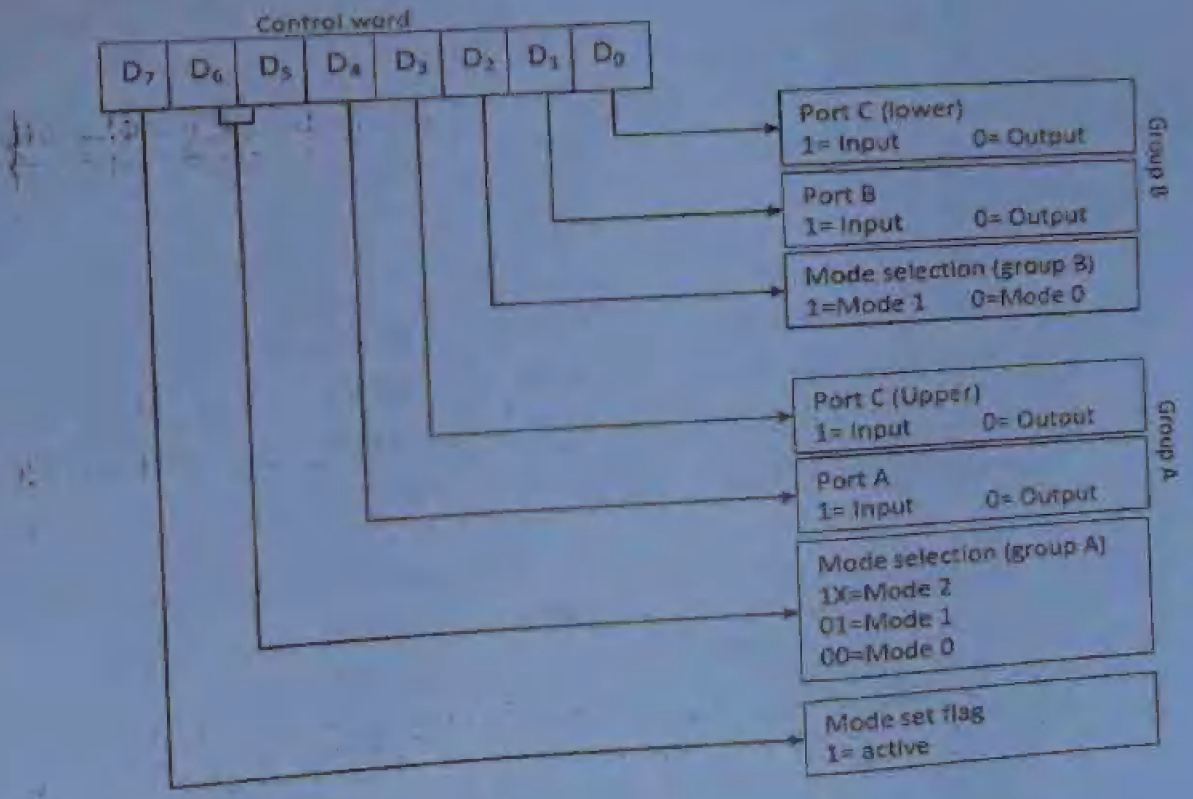
Programming the 82C55

The 8255 may be operated in one of the two modes (BSR or I/O), by initializing D7 bit in Control word register. If bit D7=1, 8255 operate in I/O mode and the bits D6-D0 determines I/O operations in various modes. If bit D7=0, 8255 operates in BSR mode and the Port C bits are used to initialize handshaking signals.

1- **I/O mode**: It is used for I/O data transfer.

2- **Bit Set-Reset mode (BSR)** : BSR mode is used to define the handshaking signals.

Control word format for I/O mode operation of PPI 8255.



Control word for the 82C55 PPI

- **If bit 7 = 1** select format for I/O mode

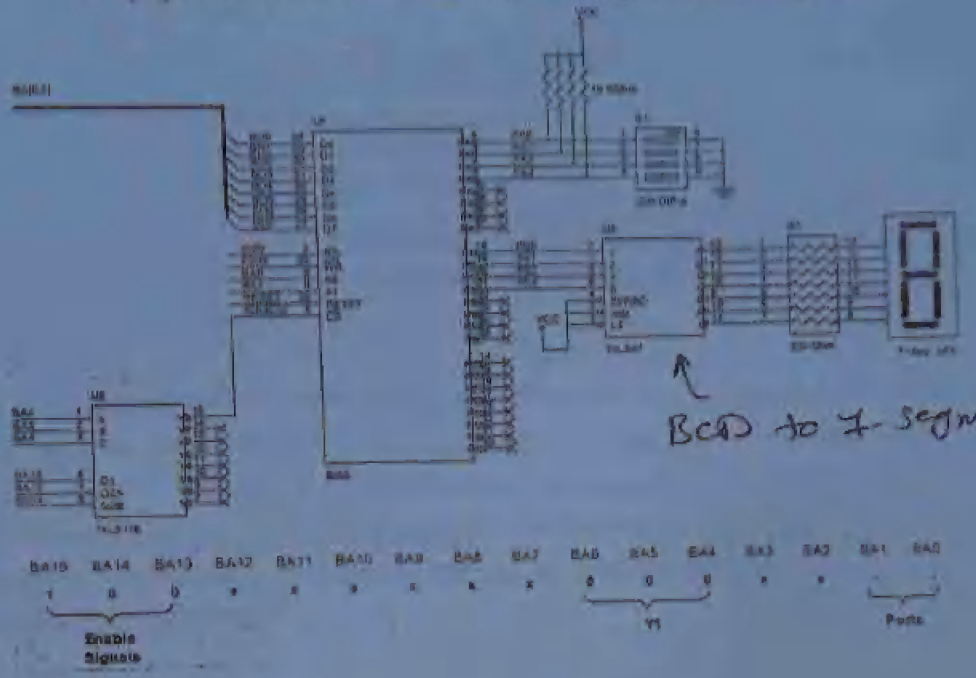
  Groups A and B either inputs or outputs /modes 0, 1, or 2

- **If bit 7 = 0** select format for Bit Set/Reset mode (BSR)

  Sets or Clears any bit of 8 bits port C (in modes 1 and 2)

(399)

Application 1: Basic Input/Output




```
PORTA   EQU     8000H
PORTB   EQU     8001H
PORTC   EQU     8002H
PCW     EQU     8003H

; 8255A Initialization
        MOV     DX, PCW          ← control word
        MOV     AL, 10011001B
        OUT     DX, AL

; To read port A unconditionally
again:  MOV     DX, PORTA
        IN      AL, DX

; To write the value to port B unconditionally
        INC     DX
        OUT     DX, AL
        JMP     again
```

Working Modes Of 8255

➢ Mode Selection

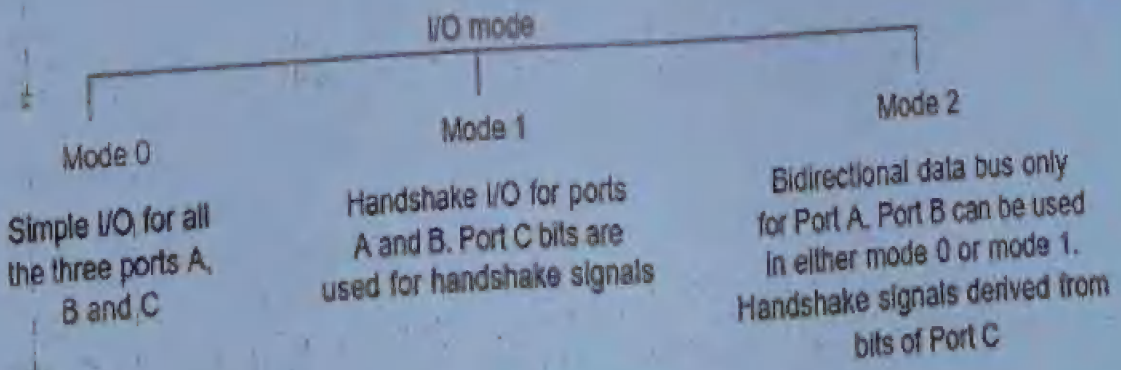
There are three basic modes of operation than can be selected by the system software:

- Mode 0 – Basic Input/Output
- Mode 1 – Strobed Input/Output
- Mode 2 – Bi-directional Bus

Mode 0 Operation It is Basic or Simple I/O. It does not use any handshake signals. It is used for interfacing an i/p device or an o/p device. It is used when timing characteristics of I/O devices is well known.

Mode 1 Operation It uses handshake I/O. 3 lines are used for handshaking. It is used for interfacing an i/p device or an o/p device. Mode 1 operation is used when timing characteristics of I/O devices is not well known, or used when I/O devices supply or receive data at irregular intervals. Handshake signals of the port inform the processor that the data is available, data transfer complete etc. More details about mode 1 operation is provided later.

Mode 2 Operation It is bi-directional handshake I/O. Mode 2 operation uses 5 lines for handshaking. It is used with an I/O device that receives data some times and sends data sometimes. Ex. Hard disk drive. Mode 2 operation is useful when timing characteristics of I/O devices is not well known, or when I/O devices supply or receive data at irregular intervals. Port A can work in Mode 0, Mode 1, or Mode 2 Port B can work in Mode 0, or Mode 1 Port C can work in Mode 0 only, if at all Port A, Port B and Port C can work in Mode 0 Port A and Port B can work in Mode 1 Only Port A can work in Mode 2



Example : What is the mode and I/O configuration for ports A, B, C of an 82C55 after its control register is loaded with 82_{16} ?

Solution:

82_{16} =

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	0

$D_7 = 1$ then mode set flag is active
$D_6\ D_5 = 00$ then Group A (A and C upper) is in mode 0
$D_4 = 0$ A is an output
$D_3 = 0$ C upper is an output
$D_2 = 0$ then Group B (B and C lower) is in mode 0
$D_1 = 1$ B is an input
$D_0 = 0$ C lower is an output

Example : Write down 82C55 control word that set Port A, Port B and Port C lower as input in mode 0, and set Port C upper as output in mode 0.

Solution:
Mode set flag is active then $D_7 = 1$
then Group A (A and C upper) is in mode 0 $D_6\ D_5 = 00$
A is aninput $D_4 = 1$
C upper is an output $D_3 = 0$
then Group B (B and C lower) is in mode 0 $D_2 = 0$
B is an input $D_1 = 1$
C lower is aninput $D_0 = 1$

Control word = 93H

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	1

Mode Definition Summary

	MODE 0	
	IN	OUT
PA_0	IN	OUT
PA_1	IN	OUT
PA_2	IN	OUT
PA_3	IN	OUT
PA_4	IN	OUT
PA_5	IN	OUT
PA_6	IN	OUT
PA_7	IN	OUT
PB_0	IN	OUT
PB_1	IN	OUT
PB_2	IN	OUT
PB_3	IN	OUT
PB_4	IN	OUT
PB_5	IN	OUT
PB_6	IN	OUT
PB_7	IN	OUT
PC_0	IN	OUT
PC_1	IN	OUT
PC_2	IN	OUT
PC_3	IN	OUT
PC_4	IN	OUT
PC_5	IN	OUT
PC_6	IN	OUT
PC_7	IN	OUT

MODE 1	
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
$INTR_B$	$INTR_B$
IBF_B	$\overline{OBF}_B$
$\overline{STB}_B$	$\overline{ACK}_B$
$INTR_A$	$INTR_A$
$\overline{STB}_A$	I/O
IBF_A	I/O
I/O	$\overline{ACK}_A$
I/O	$\overline{OBF}_A$

MODE 2
GROUP A ONLY
⟷
⟷
⟷
⟷
⟷
⟷
⟷
⟷
—
—
—
—
—
—
—
—
I/O
I/O
I/O
$INTR_A$
$\overline{STB}_A$
IBF_A
$\overline{ACK}_A$
$\overline{OBF}_A$

MODE 0 OR MODE 1 ONLY

Write down the mode 0 control words for the following two cases:

(a) Port A = Input port, Port B = not used, Port C_U = Input port and Port C_L = Output port.

(b) Port A = Output port, Port B = Input port, Port C = Output port

Ans: The control words for the two cases will be as follows:

(a)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
1	0	0	1	1	0	0	0	=	98_H
I/O mode	Mode 0 for Port A		Port A input	Port C_U input	Port B not used		Port C_L output		

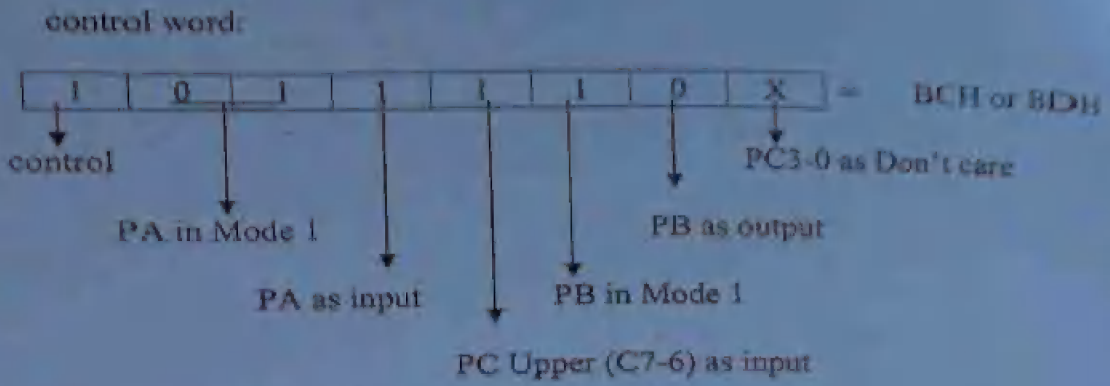
Thus, the control word would be = 98_H

(b)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
1	0	0	0	0	0	1	0	=	82_H
I/O mode	Mode 0 for Port A		Port A output	Port C_U output	Mode 0 for Port B	Port B input	Port C_L output		

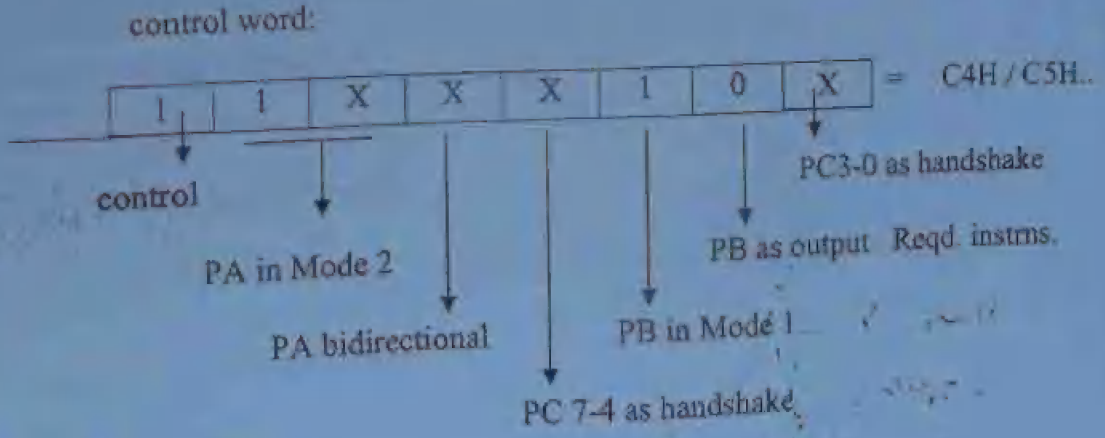
Thus, the control word would be = 82_H.

Ex. 2 Configure Port A as input in Mode 1, Port B as output in mode 1, Port C7-6 as input ports (PC5-0 are handshake lines, some are input lines and others are output, So they are shown as X)



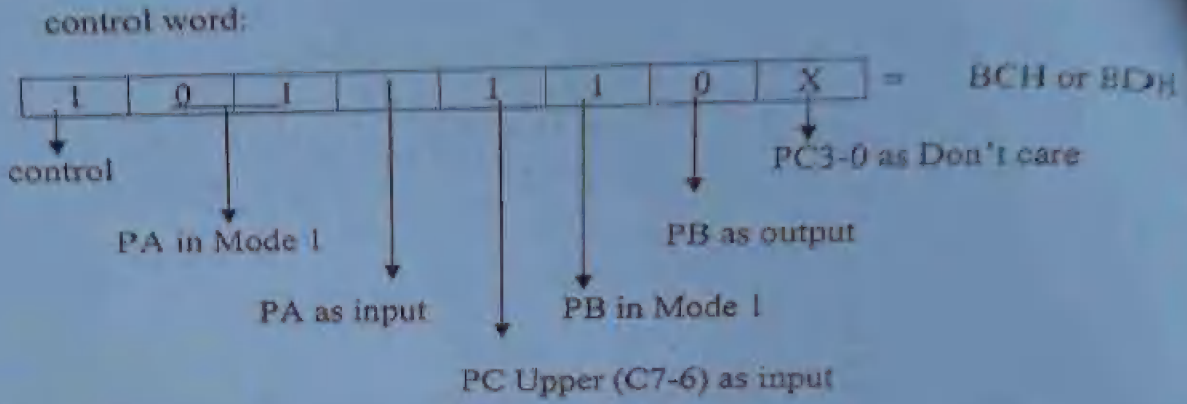
Required program segment for the configuration:
MOV AL, BCH
OUT 7FH, AL

Ex, 3:Configure Port A in Mode 2, Port B as output in mode 1. (PC7-3 are handshake lines for Port A and PC2-0 are handshake signals for port B)



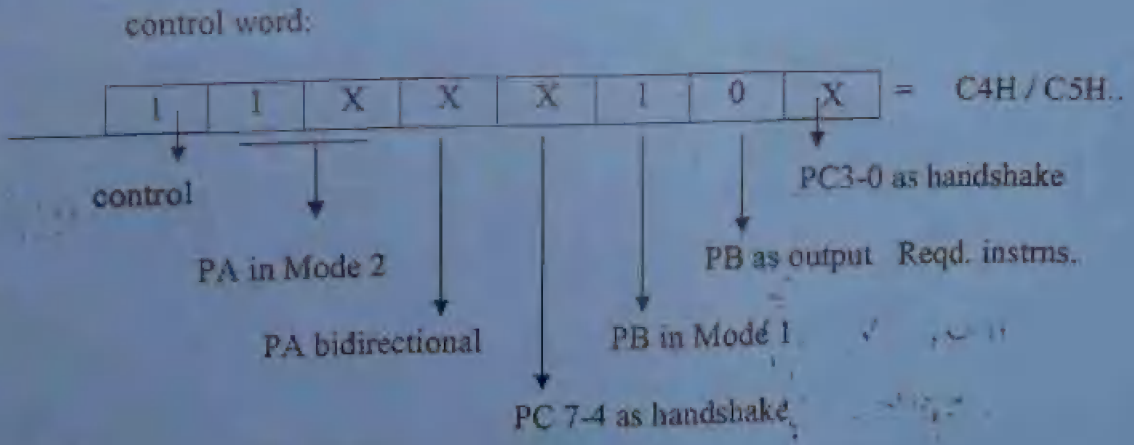
Required program segment for the configuration:
MOV AL, C4H
OUT 7FH, AL

Ex. 2: Configure Port A as input in Mode 1, Port B as output in mode 1, Port C7-6 as input ports. (PC5-0 are handshake lines, some are input lines and others are output. So they are shown as X)



Required program segment for the configuration:
MOV AL, BCH
OUT 7FH, AL

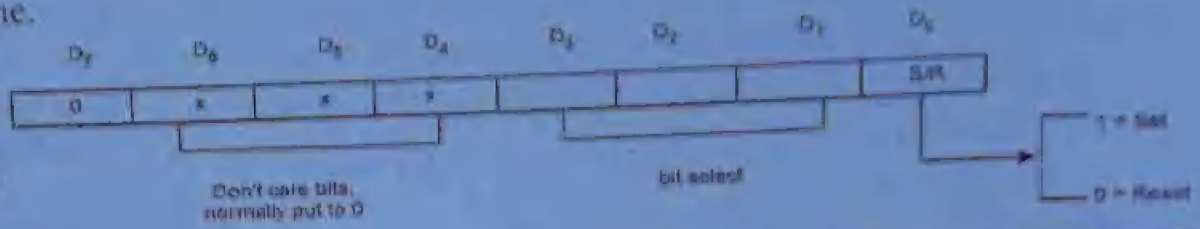
Ex, 3:Configure Port A in Mode 2, Port B as output in mode 1. (PC7-3 are handshake lines for Port A and PC2-0 are handshake signals for port B)



Required program segment for the configuration:
MOV AL, C4H
OUT 7FH, AL

Control word format in the BSR mode

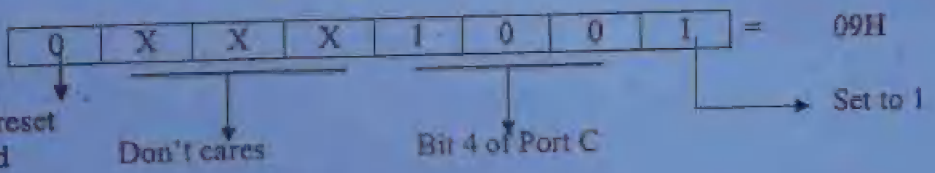
The content of the control word register will be as follows, when used in the BSR mode and selects (either Sets or Resets) a particular bit of Port C at a time.



D_3	D_2	D_1	Particular bit of Port C selected
0	0	0	Bit 0
0	0	1	Bit 1
0	1	0	Bit 2
0	1	1	Bit 3
1	0	0	Bit 4
1	0	1	Bit 5
1	1	0	Bit 6
1	1	1	Bit 7

The CWR in the BSR mode

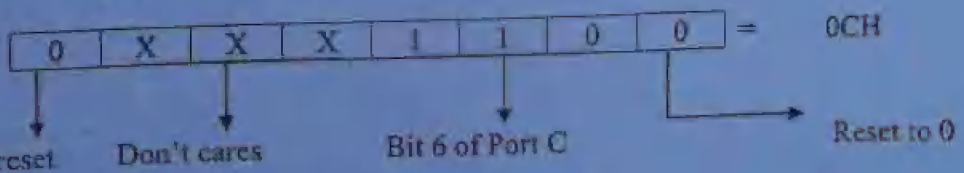
Ex. 1: Set to 1 bit 4 of Port C



Required program segment for setting bit 4 of Port C:

```
MOV AL, 09H
OUT 7FH, AL
```

Ex. 2: Reset to 0 bit 6 of Port C



Required program segment for resetting bit 6 of Port C:

```
MOV AL, 0CH
OUT 7FH, AL
```

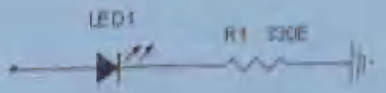
Q- Write a BSR control word to set bits PC7 and PC0 and to reset them after 1 second delay?

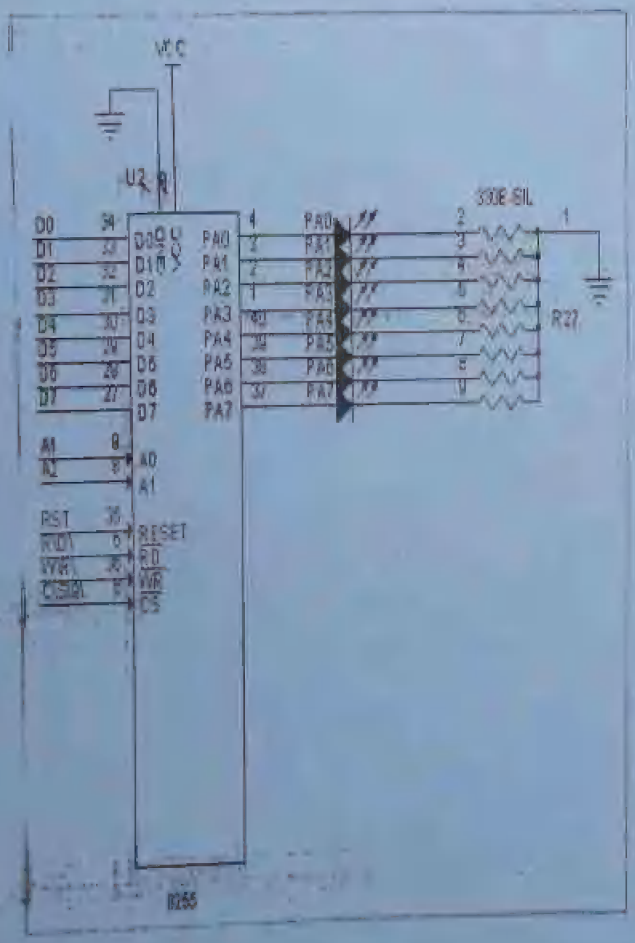
To set or reset any particular bit of Port C in the BSR mode, the control word register is to be appropriately loaded. The above is done by loading the accumulator and sending the same to the control register (i.e., by sending the same to the address of the control register). The address of control word register (CWR) is 83H (10000011).

Program:

```
MOV   AL, 0FH     (Accumulator loaded with 0FH to set PC7 bit of Port C)
OUT   83,AL       (This sets PC7 bit of  Port C)
MOV   AL, 01H     (Accumulator loaded with 01H to set PC0 bit of Port C)
OUT   83,AL       (This sets PC0 bit of  Port C)
CALL DELAY        (Assume the DELAY is for 1 second)
MOV AL, 00H       (Accumulator loaded with 00 H  to reset PC0 bit of Port C)
OUT 83,AL         (This resets PC0 bit of Port C)
MOVA, 0EH         (Accumulator loaded with 0E H  to reset PC7 bit of Port C)
OUT 83,AL         (This resets PC7 bit of Port C)
```

PIN ASSIGNMENT WITH 8086

	Board LEDs	8255 Lines	LED Selection
DIGITAL OUTPUTS	LD1	PA.0	 Make Pin High – LED ON Make Pin Low – LED OFF
	LD2	PA.1	
	LD3	PA.2	
	LD4	PA.3	
	LD5	PA.4	
	LD6	PA.5	
	LD7	PA.6	
	LD8	PA.7	

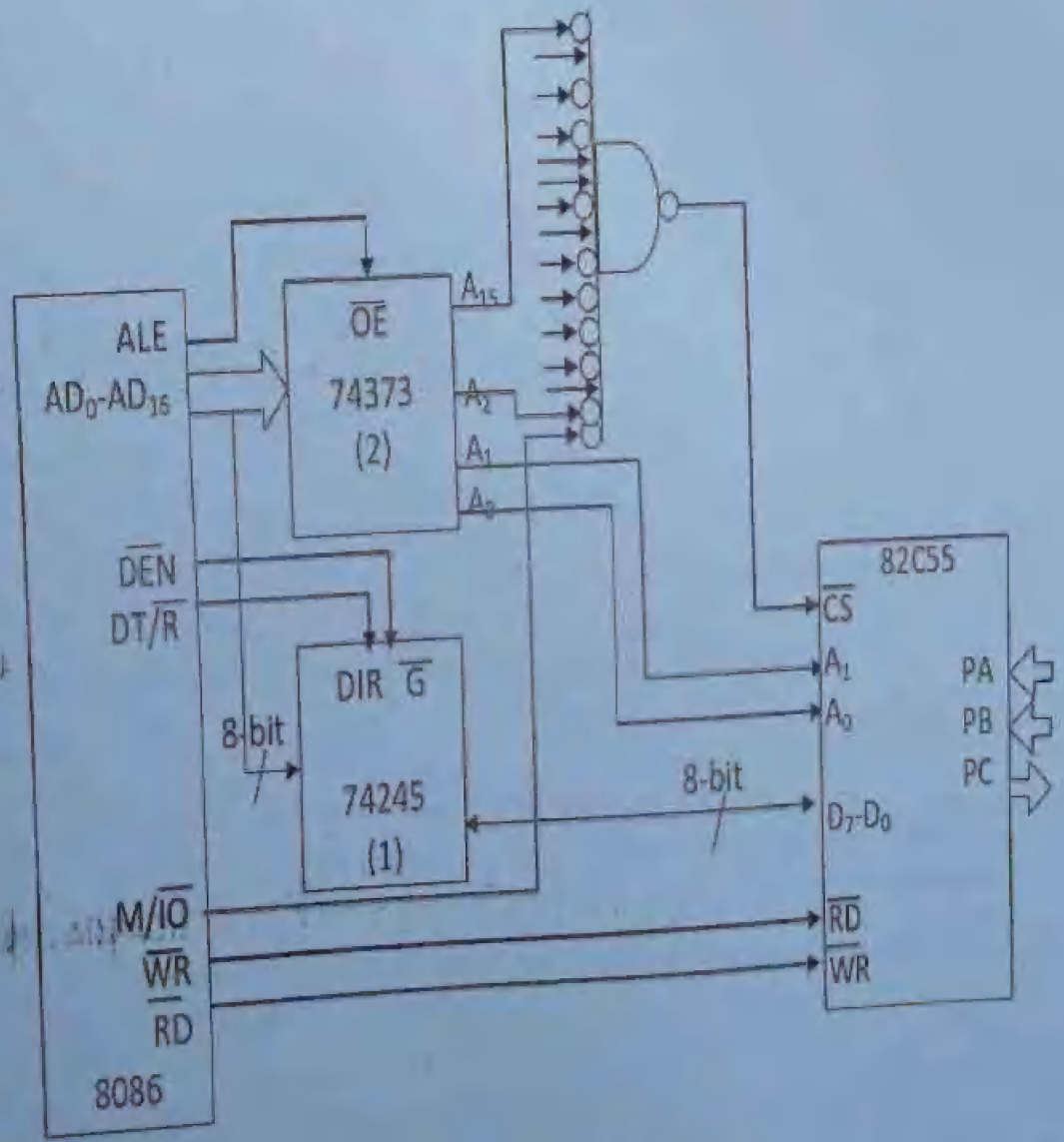



```
MOV AL, 80
MOV DX, FF36
OUT DX, AL
BEGIN: MOV AL, 00
MOV DX, FF30
OUT DX, AL
CALL DELAY
MOV AL, FF
OUT DX, AL
CALL DELAY
JMP BEGIN
DELAY: MOV CX, FFFF
P0: DEC CX
JNE P0
RET
```

Example: In 8086's 8-bit isolated I/O system, an 82C55 PPI is connected so that the address of A, B, C ports, and Control register are $4D08_{16}$, $4D09_{16}$, $4D0A_{16}$ and$4D0B_{16}$ respectively.

a- Draw the hardware interface circuit.

b- Write program to set Register A, B as input and Register C as output (all in mode 0). Then continuously receive two unsigned number from Registers A and B, compare them and output the larger to Register C.



Control word =92H

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	0	1	0

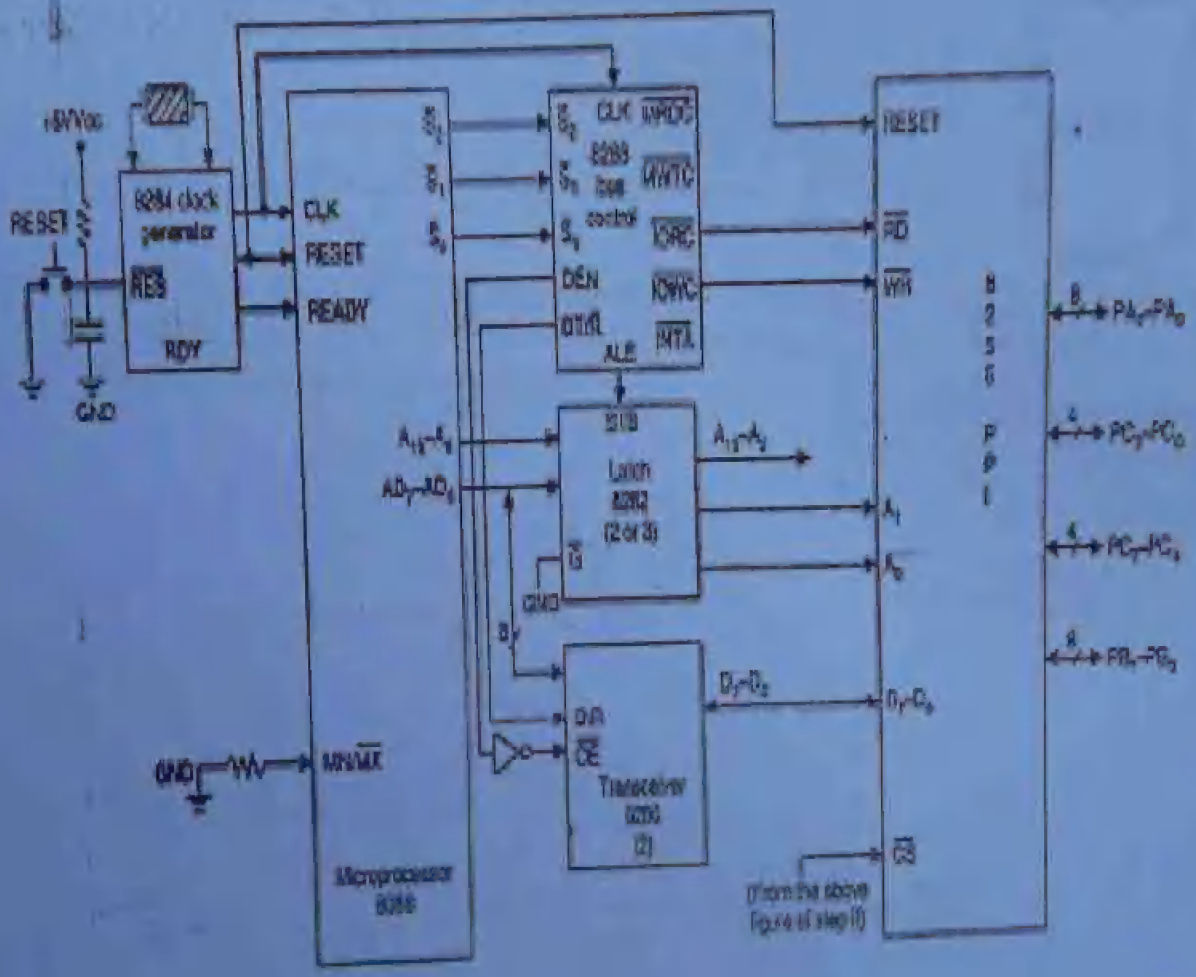
```
                MOV AL, 92H
                MOV DX, 4D0BH
                OUT DX, AL
again:          MOV DL, 08H  (because DH is the same)
                IN AL,DX
                MOV BL, AL
                INC DL
                IN AL, DX
                CMP AL, BL
                JNC no_exchange
                MOV AL, BL
no_exchange:    INC DL
                OUT DX, AL
                MOV CX, FFFFH
delayloop:      DEC CX
                JNZ delayloop
                JMP again
```

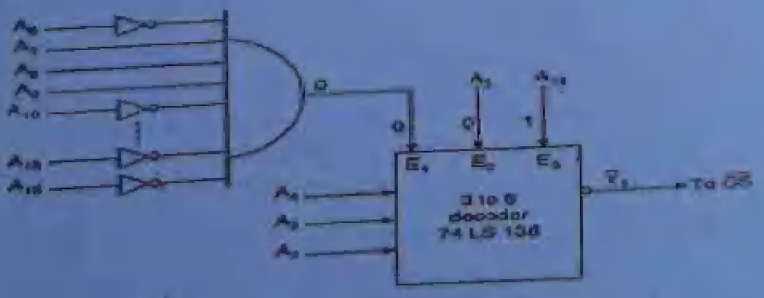
Example : Repeat the last example using memory mapped I/O.

Hint:

- **Isolated I/O: no need to decode all address lines.**
  - **fixed port # : A0-A7 must be decoded encoded**
  - **variable port # : A0-A15 must be decoded encoded**
- **Memory mapped I/O : all address line must be encoded (A0-A19)**

Ex: Interface an 8255 with 8088 in the maximum mode configuration from address 4384h .





4384H=0100001110000100=$A_{15}A_{14}A_{13}\ A_{12}\ A_{11}\ A_{10}A_9A_8\ A_7\ A_6\ A_5\ A_4\ A_3A_2A_1\ A_0$

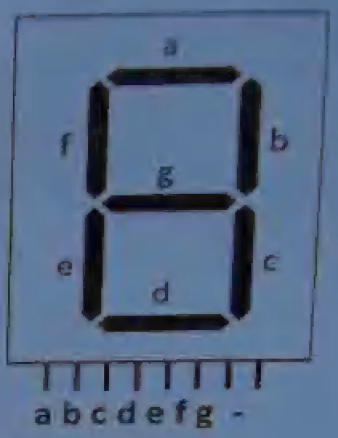
Seven-segment display Interface

Seven-segment display used to display number (and letters). Seven segment display labeling is shown in figure below.

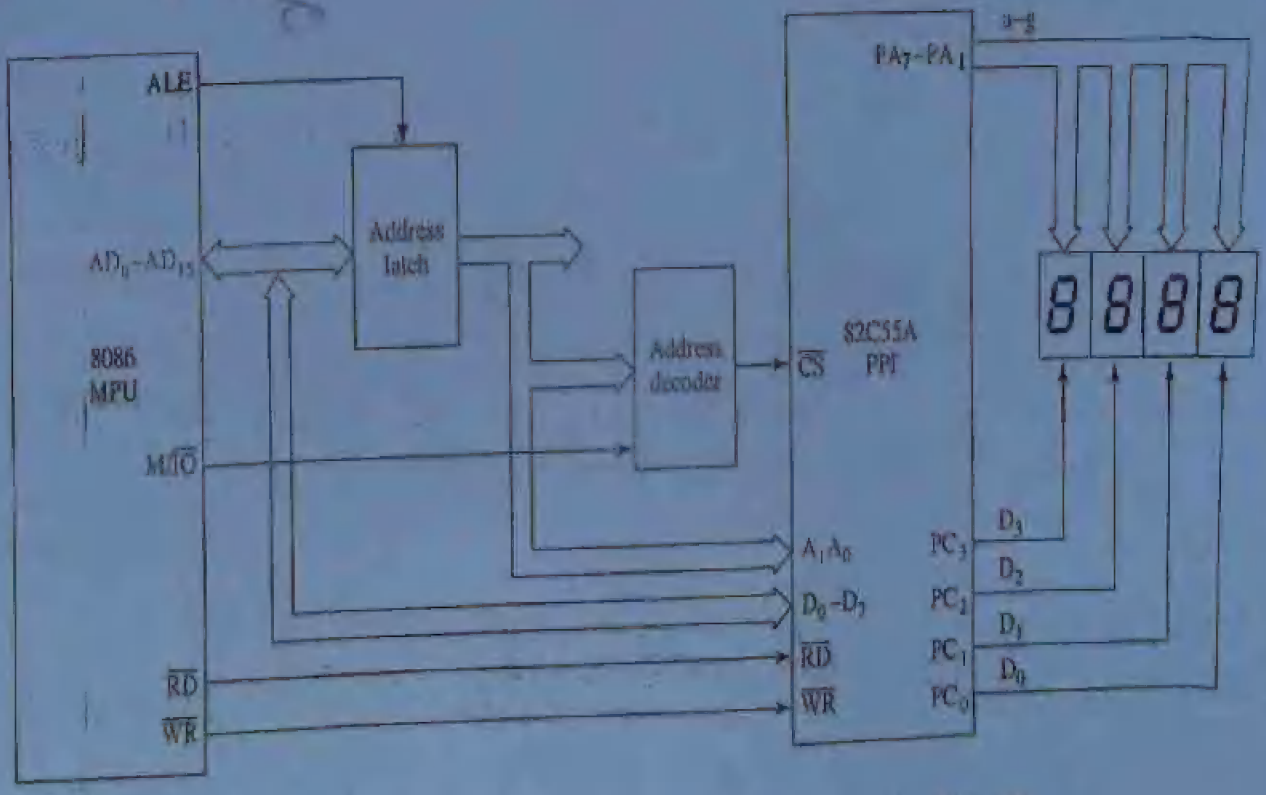
For example to display number (9)
the bit must set as follow:

a	b	c	d	e	f	g	
1	1	1	1	0	1	1	x

If the byte **F6**16 (or **F7** 16 because bit D_0 is neglected) is written to address of the seven-segment display then it will display number nine.



Seven-segment display



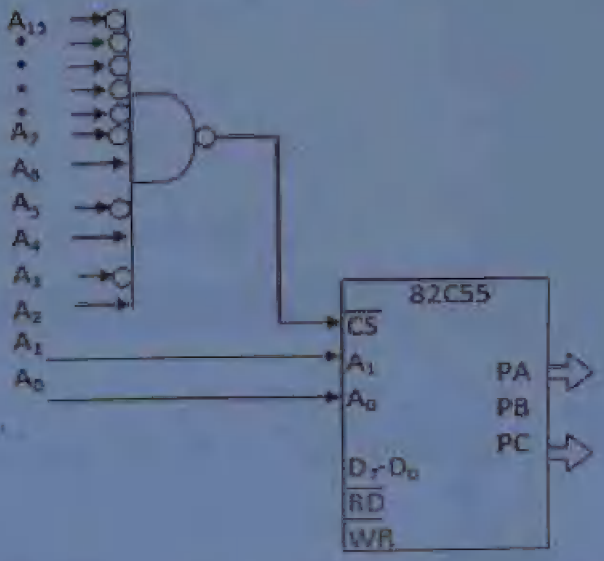
Four –digit Seven-segment display interface to 8086 microprocessor using 82C55 PPI

Example

The above Figure shows an interface of **four-digit seven-segment** numeric display. The circuit use Port A to output value to the display, and Port C lower to select which digit is active.

a) Draw the detail of the **address decode** circuit that make the address of port A is 0054_{16}, address of port C is 0056_{16} and address of control register is 0057

b) Write program to display the word **PASS**. Note that the program must continuously set each one of the four digits because they are multiplexed.



Control word =80H

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	X	0	X	0

Pcharacter : CE H

a	b	c	d	e	f	g	
1	1	0	0	1	1	1	x

Acharacter : EE H

a	b	c	d	e	f	g	
1	1	1	0	1	1	1	x

Scharacter :B6 H

a	b	c	d	e	f	g	
1	0	1	1	0	1	1	x

Creating time delay using instructions

Instruction	Time of execution
MOV Reg8/16	4 Clock
DEC Reg 8/16	3 Clock
JNZ/JZ	16/4 Clock
LOOP	17/5 Clock

1- Generate short time delay

```
        MOV CX ,N
again:  LOOP  again
```

The maximum value of N = FFFFH=63535

$$\text{Time delay} \cong 17\ N\ .\ T\ \text{period}$$

Example

If the system frequency = 5MHz. What is the time delay generated by the following routine.

```
        MOV CX,28000
again : LOOP  again
```

$$\text{Time delay} \cong 17\ N\ .\ T\ \text{period} \cong 28000\ .\ 17\ .\frac{1}{5\times10^6} \cong 95.2\ \text{ms}$$

Example

Generate time delay 95.2 ms If the system frequency = 5MHz.

$$\text{Time delay} \cong 17\ N\ .\ T\ \text{period} \cong N\ .\ 17\ .\frac{1}{5\times10^6} \cong 95.2\ \text{ms}$$

$$N = 28000$$

```
        MOV CX,28000
again : LOOP  again
```

2- Generate long time delay

Program

```
        MOV AL, 80H
        OUT 57H, AL    (set the control register)

again:  MOV AL, 08H
        OUT 56H, AL (enable fourth digit)
        MOV AL, CEH
        OUT 54H, AL (display character P)

        MOV AL, 04H
        OUT 56H, AL (enable third digit)
        MOV AL, EEH
        OUT 54H, AL (display character A)

        MOV AL, 02H
        OUT 56H, AL (enable second digit)
        MOV AL, B6H
        OUT 54H, AL (display character S)

        MOV AL, 01H
        OUT 56H, AL (enable first digit)
        MOV AL, B6H
        OUT 54H, AL (display character S)

        JMP again
```

Example:

Repeat the last example using memory-mapped IO.

```
        MOV DX ,N1      → 4T
again2: MOV CX ,N2      → 4T
again1: LOOP again1     → 17 N2T
        DEC DX          → 3T
        JNZ again2      → 16T
```

Time delay ≅ [4T+17N$_2$T+3T+16T] N$_1$

Time delay ≅ [17N$_2$ N$_1$T+23 N$_1$ T]

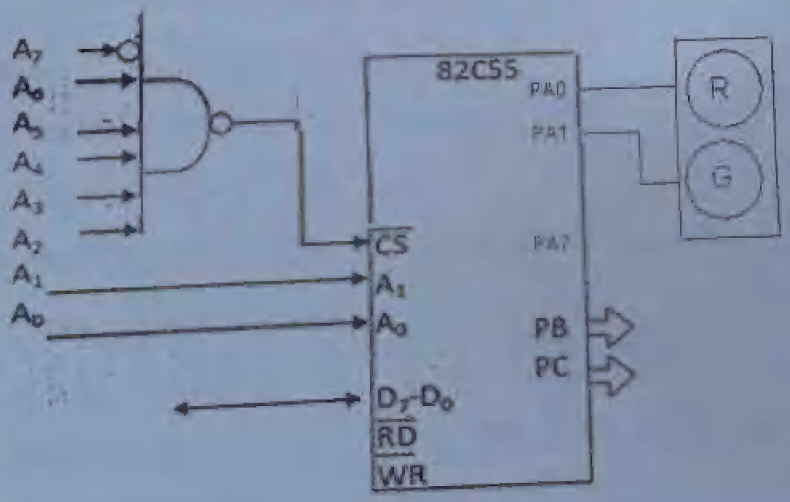
Example

For the microprocessor based system shown in figure below. Write the instructions required to program the traffic light as follow:-

Red (ON) , Green (OFF) for 23.6 Sec.

Red (OFF) , Green (ON) for 23.6 Sec.

Assume the system frequency 4.7 MHz.



Port Address

CS' A1 A0	Selection		
0 1 1 1 1 1 0 0	Port A	PA	=7CH
0 1 1 1 1 1 0 1	Port B	PA	=7DH
0 1 1 1 1 1 1 0	Port C	PA	=7EH
0 1 1 1 1 1 1 1	CWR	CWR	=7FH

Control word =80H

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

Delay:

```
        MOV DX,N1
again2: MOV CX,N2
again1: LOOP again1
        DEC DX
        JNZ again2
        RET
```

Time delay $\cong 23.6 \cong [17N_2 N_1T+23 N_1 T]$,

assume N2 = **FFFFH**= $(65535)_{10}$ $\rightarrow N_1 \cong 100$

Program

```
        MOV Al ,80H          ; configuration
        OUT 7FH,AL
gain3: MOV Al,00000001B      ; Red ON   Green OFF
       OUT 7CH,AL
       CALL delay
       MOV Al,00000010B      ; Red OFF  Green ON
       OUT 7CH,AL
       CALL delay
       JMP again3
```

October 1986
Revised March 2000

FAIRCHILD SEMICONDUCTOR TM

DM74LS47
BCD to 7-Segment Decoder/Driver with Open-Collector Outputs

General Description

The DM74LS47 accepts four lines of BCD (8421) input data, generates their complements internally and decodes the data with seven AND/OR gates having open-collector outputs to drive indicator segments directly. Each segment output is guaranteed to sink 24 mA in the ON (LOW) state and withstand 15V in the OFF (HIGH) state with a maximum leakage current of 250 µA. Auxiliary inputs provided blanking, lamp test and cascadable zero-suppression functions.

Features

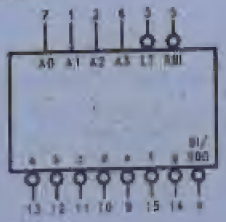
- Open-collector outputs
- Drive indicator segments directly
- Cascadable zero-suppression capability
- Lamp test input

Ordering Code:

Order Number	Package Number	Package Description
DM74LS47M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS47N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

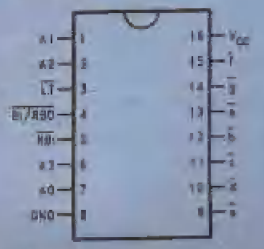
Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Logic Symbol



V_{CC} = Pin 16
GND = Pin 8

Connection Diagram



Pin Descriptions

Pin Names	Description
A0–A3	BCD Inputs
$\overline{RBI}$	Ripple Blanking Input (Active LOW)
$\overline{LT}$	Lamp Test Input (Active LOW)
$\overline{BI/RBO}$	Blanking Input (Active LOW) or Ripple Blanking Output (Active LOW)
$\overline{a}$ –$\overline{g}$	Segment Outputs (Active LOW) (Note 1)

Note 1: OC—Open Collector

Truth Table

Decimal or Function	Inputs							Outputs							Note
	$\overline{LT}$	$\overline{RBI}$	A3	A2	A1	A0	$\overline{BI}/\overline{RBO}$	a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	(Note 2)
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H	(Note 2)
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L	
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L	
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L	
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L	
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L	
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H	
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L	
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L	
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L	
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L	
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L	
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L	
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L	
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H	
$\overline{BI}$	X	X	X	X	X	X	L	H	H	H	H	H	H	H	(Note 3)
$\overline{RBI}$	H	L	L	L	L	L	L	H	H	H	H	H	H	H	(Note 4)
$\overline{LT}$	L	X	X	X	X	X	H	L	L	L	L	L	L	L	(Note 5)

Note 2: $\overline{BI}/\overline{RBO}$ is wire-AND logic serving as blanking input ($\overline{BI}$) and/or ripple-blanking output ($\overline{RBO}$). The blanking out ($\overline{BI}$) must be open or held at a HIGH level when output functions 0 through 15 are desired, and ripple-blanking input ($\overline{RBI}$) must be open or at a HIGH level if blanking of a decimal 0 is not desired. X = input may be HIGH or LOW.

Note 3: When a LOW level is applied to the blanking input (forced condition) all segment outputs go to a HIGH level regardless of the state of any other input condition.

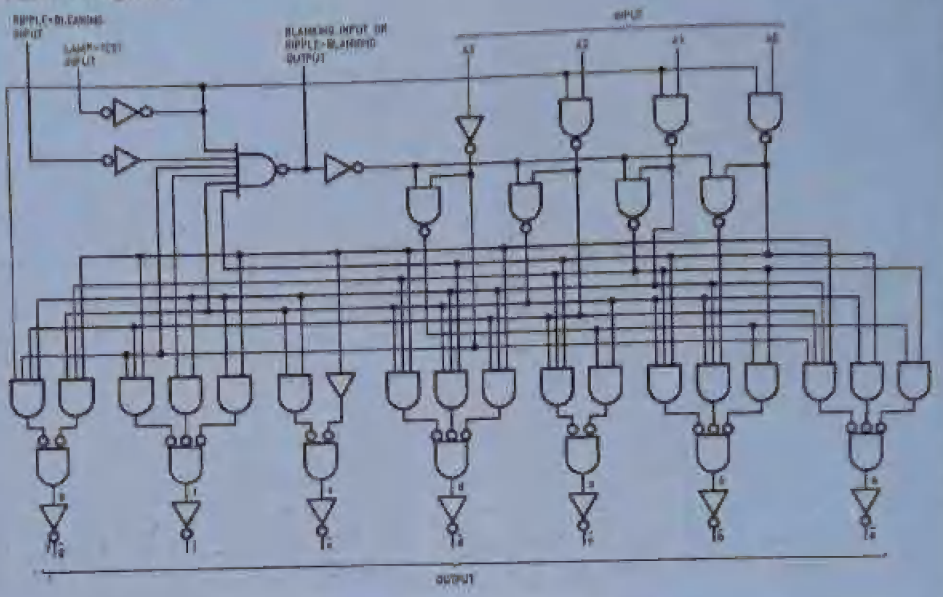
Note 4: When ripple-blanking input ($\overline{RBI}$) and inputs A0, A1, A2 and A3 are LOW level, with the lamp test input at HIGH level, all segment outputs go to a HIGH level and the ripple-blanking output ($\overline{RBO}$) goes to a LOW level (response condition).

Note 5: When the blanking input/ripple-blanking output ($\overline{BI}/\overline{RBO}$) is OPEN or held at a HIGH level, and a LOW level is applied to lamp test input, all segment outputs go to a LOW level.

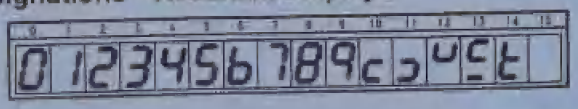
Functional Description

The DM74LS47 decodes the input data in the pattern indicated in the Truth Table and the segment identification illustration. If the input data is decimal zero, a LOW signal applied to the $\overline{RBI}$ blanks the display and causes a multi-digit display. For example, by grounding the $\overline{RBI}$ of the highest order decoder and connecting its $\overline{BI}/\overline{RBO}$ to $\overline{RBI}$ of the next lowest order decoder, etc., leading zeros will be suppressed. Similarly, by grounding $\overline{RBI}$ of the lowest order decoder and connecting its $\overline{BI}/\overline{RBO}$ to $\overline{RBI}$ of the next highest order decoder, etc., trailing zeros will be suppressed. Leading and trailing zeros can be suppressed simultaneously by using external gates, i.e., by driving $\overline{RBI}$ of a intermediate decoder from an OR gate whose inputs are $\overline{BI}/\overline{RBO}$ of the next highest and lowest order decoders. $\overline{BI}/\overline{RBO}$ also serves as an unconditional blanking input. The internal NAND gate that generates the $\overline{RBO}$ signal has a resistive pull-up, as opposed to a totem pole, and thus $\overline{BI}/\overline{RBO}$ can be forced LOW by external means, using wired-collector logic. A LOW signal thus applied to $\overline{BI}/\overline{RBO}$ turns off all segment outputs. This blanking feature can be used to control display intensity by varying the duty cycle of the blanking signal. A LOW signal applied to $\overline{LT}$ turns on all segment outputs, provided that $\overline{BI}/\overline{RBO}$ is not forced LOW.

Logic Diagram



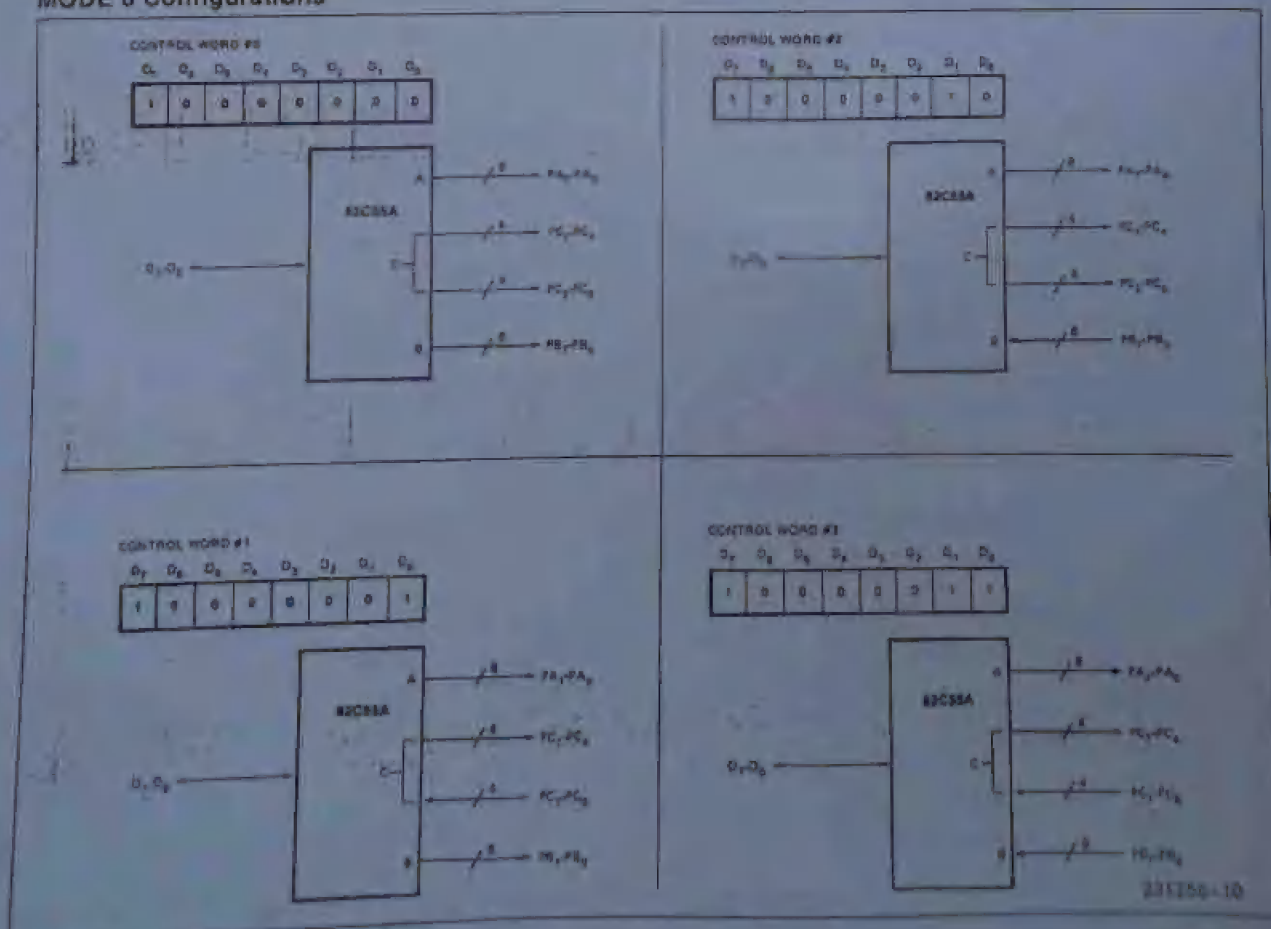
Numerical Designations—Resultant Displays



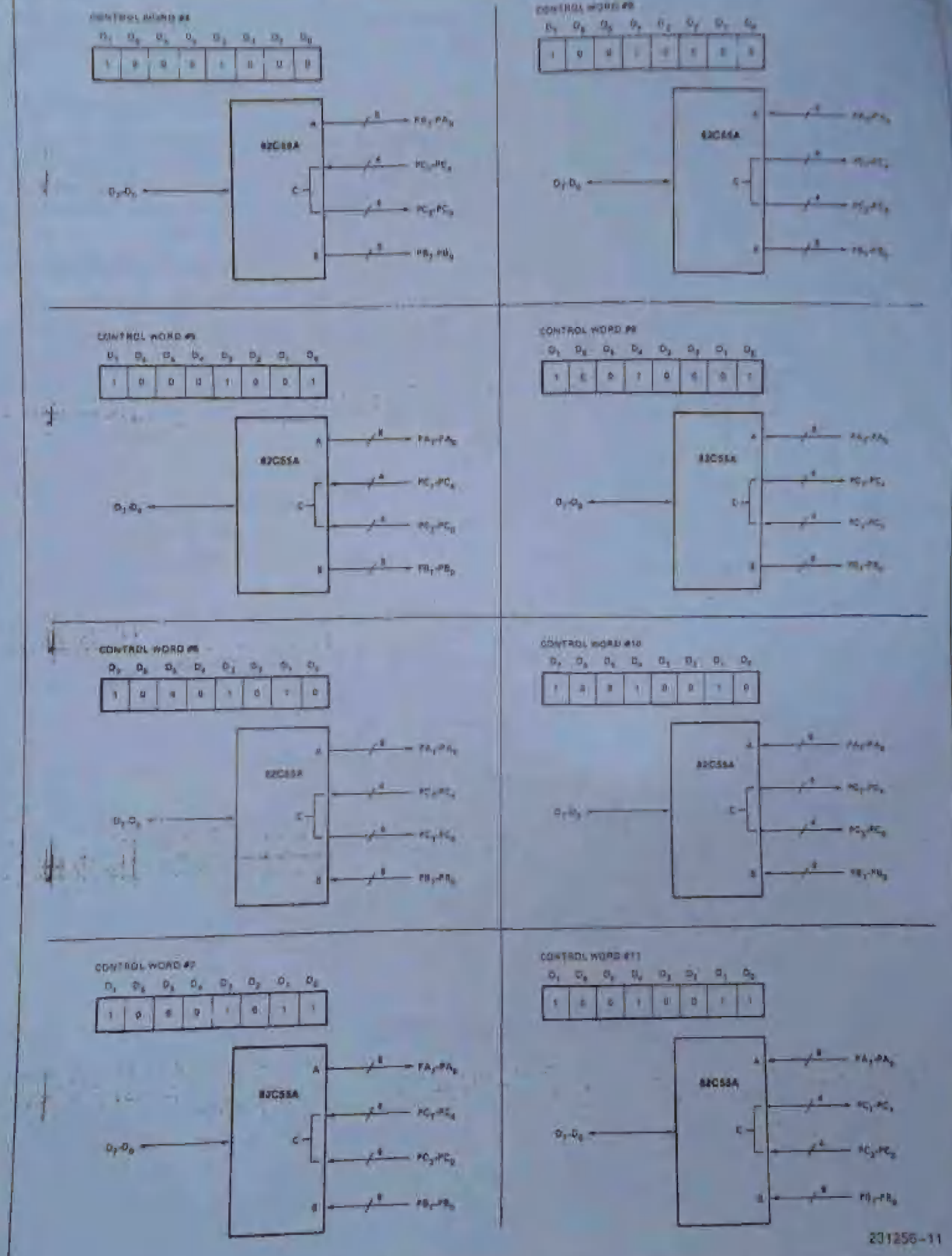
MODE 0 Port Definition

A		B		GROUP A			GROUP B	
D_4	D_3	D_1	D_0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 Configurations



MODE 0 Configurations (Continued)



231256-11

Input Control Signal Definition

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by $\overline{STB}$ input being low and is reset by the rising edge of the $\overline{RD}$ input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

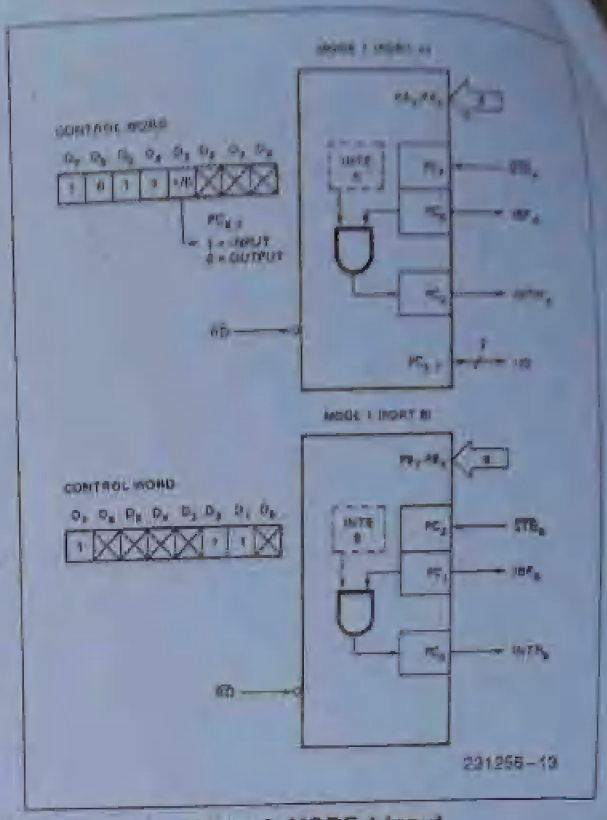


Figure 8. MODE 1 Input

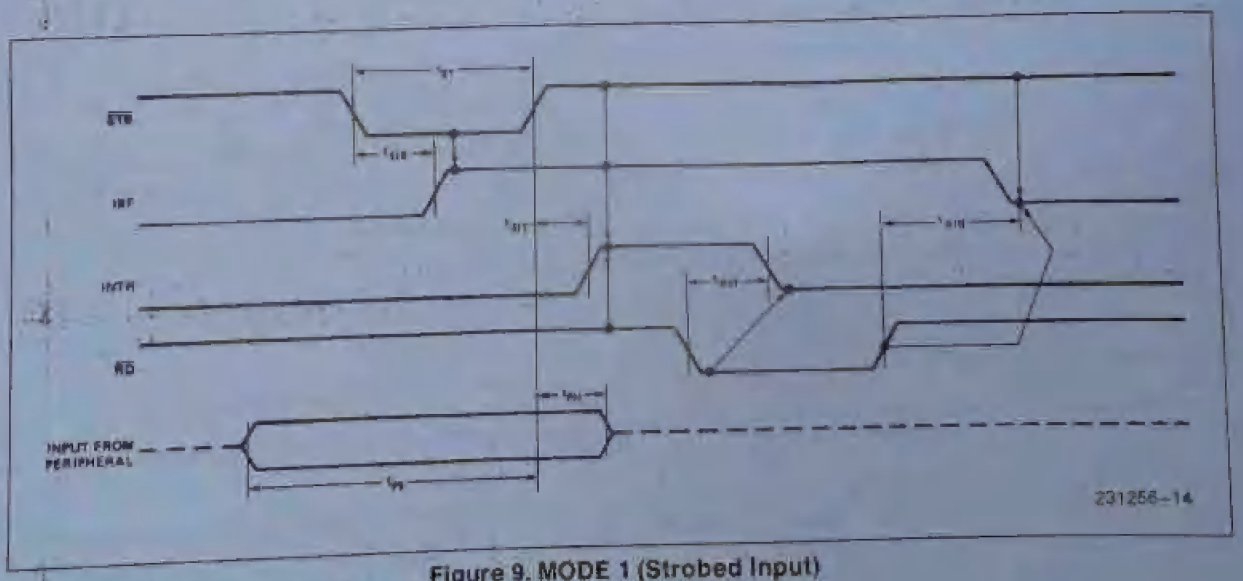


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 82C55A that the data from Port A or Port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", $\overline{OBF}$ is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

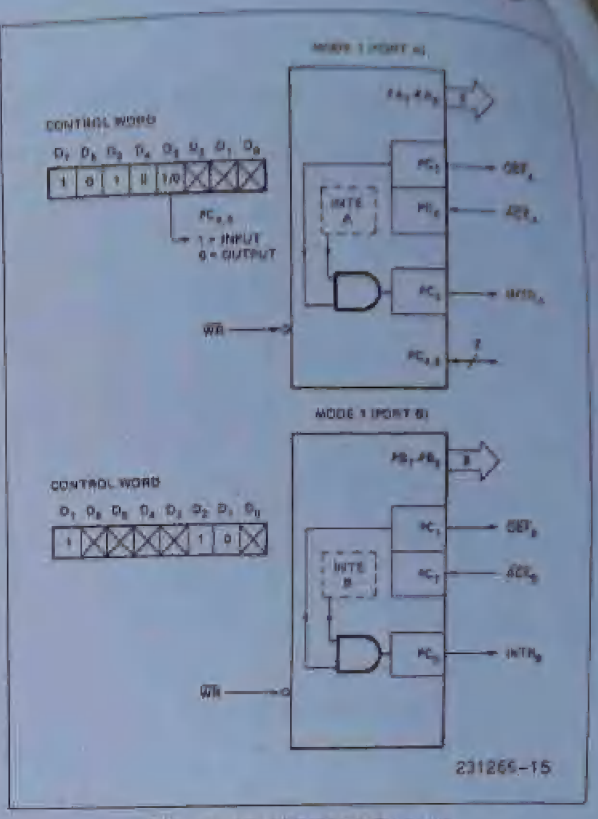


Figure 10. MODE 1 Output

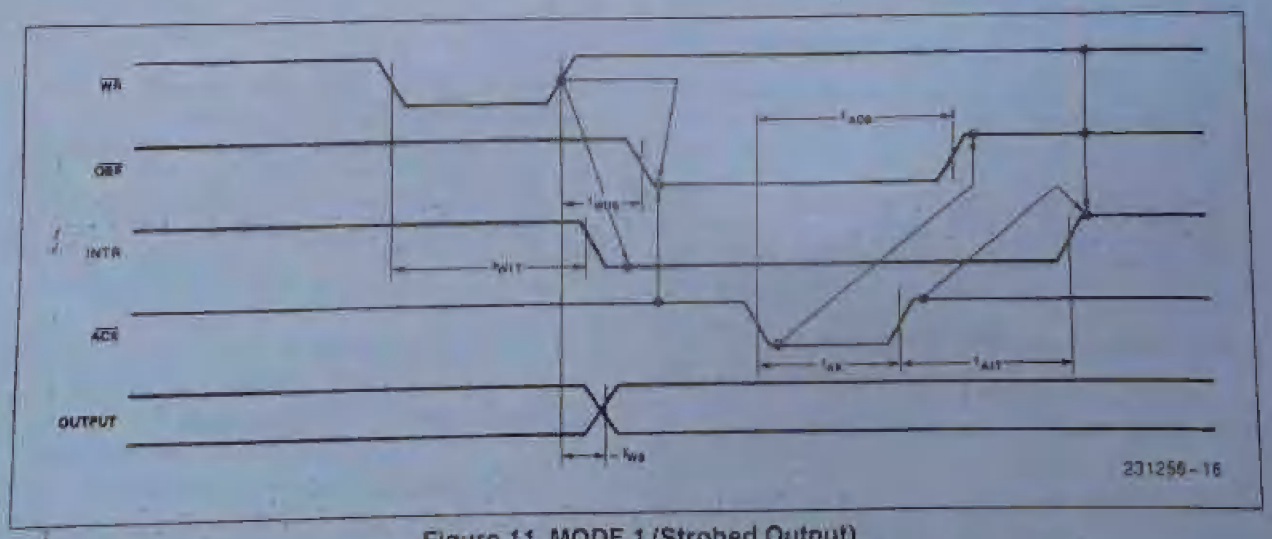


Figure 11. MODE 1 (Strobed Output)

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

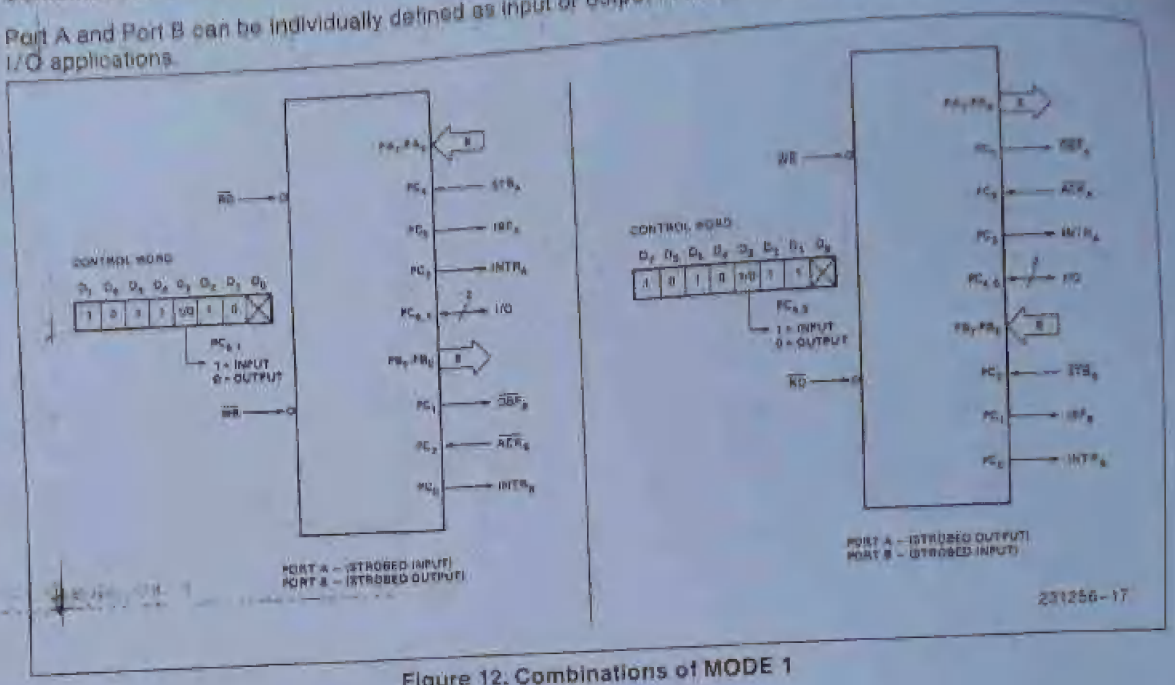


Figure 12. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A **only**.
- One 8-bit, bi-directional bus port (Port A) and a 5-bit control port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to port A.

$\overline{ACK}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of Port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with $\overline{OBF}$). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

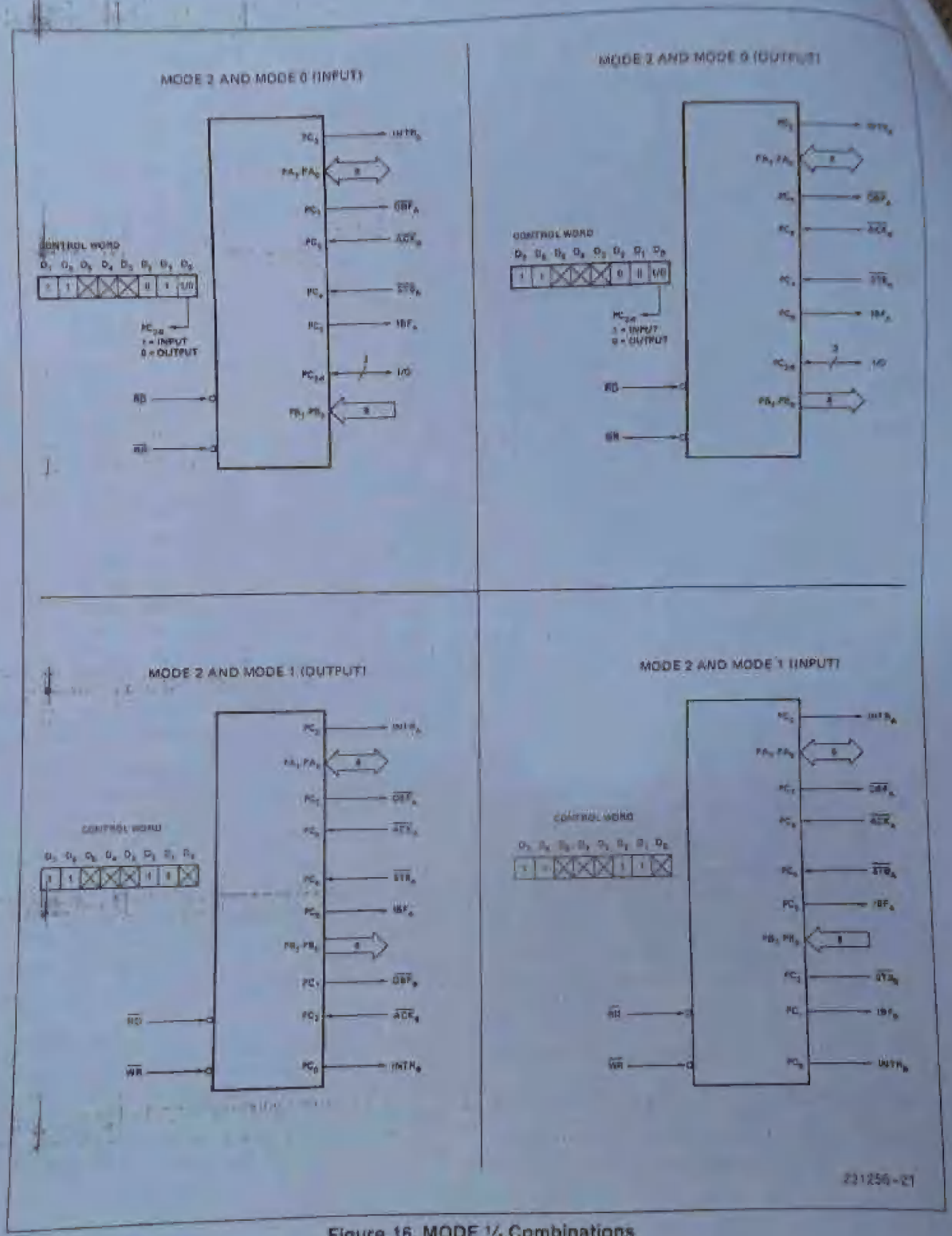


Figure 16. MODE ¼ Combinations

Mode Definition Summary

	MODE 0	
	IN	OUT
PA_0	IN	OUT
PA_1	IN	OUT
PA_2	IN	OUT
PA_3	IN	OUT
PA_4	IN	OUT
PA_5	IN	OUT
PA_6	IN	OUT
PA_7	IN	OUT
PB_0	IN	OUT
PB_1	IN	OUT
PB_2	IN	OUT
PB_3	IN	OUT
PB_4	IN	OUT
PB_5	IN	OUT
PB_6	IN	OUT
PB_7	IN	OUT
PC_0	IN	OUT
PC_1	IN	OUT
PC_2	IN	OUT
PC_3	IN	OUT
PC_4	IN	OUT
PC_5	IN	OUT
PC_6	IN	OUT
PC_7	IN	OUT

MODE 1	
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
$INTR_B$	$INTR_B$
IBF_B	$\overline{OBF}_B$
$\overline{STB}_B$	$\overline{ACK}_B$
$INTR_A$	$INTR_A$
$\overline{STB}_A$	I/O
IBF_A	I/O
I/O	$\overline{ACK}_A$
I/O	$\overline{OBF}_A$

MODE 2
GROUP A ONLY
↔
↔
↔
↔
↔
↔
↔
↔
—
—
—
—
—
—
—
—
I/O
I/O
I/O
$INTR_A$
$\overline{STB}_A$
IBF_A
$\overline{ACK}_A$
$\overline{OBF}_A$

MODE 0 OR MODE 1 ONLY

Special Mode Combination Considerations

There are several combinations of modes possible. For any combination, some or all of the Port C lines are used for control or status. The remaining bits are either inputs or outputs as defined by a "Set Mode" command.

During a read of Port C, the state of all the Port C lines, except the $\overline{ACK}$ and $\overline{STB}$ lines, will be placed on the data bus. In place of the $\overline{ACK}$ and $\overline{STB}$ line states, flag status will appear on the data bus in the PC2, PC4, and PC6 bit positions as illustrated by Figure 18.

Through a "Write Port C" command, only the Port C pins programmed as outputs in a Mode 0 group can be written. No other pins can be affected by a "Write Port C" command, nor can the interrupt enable flags be accessed. To write to any Port C output programmed as an output in a Mode 1 group or to change an interrupt enable flag, the "Set/Reset Port C Bit" command must be used.

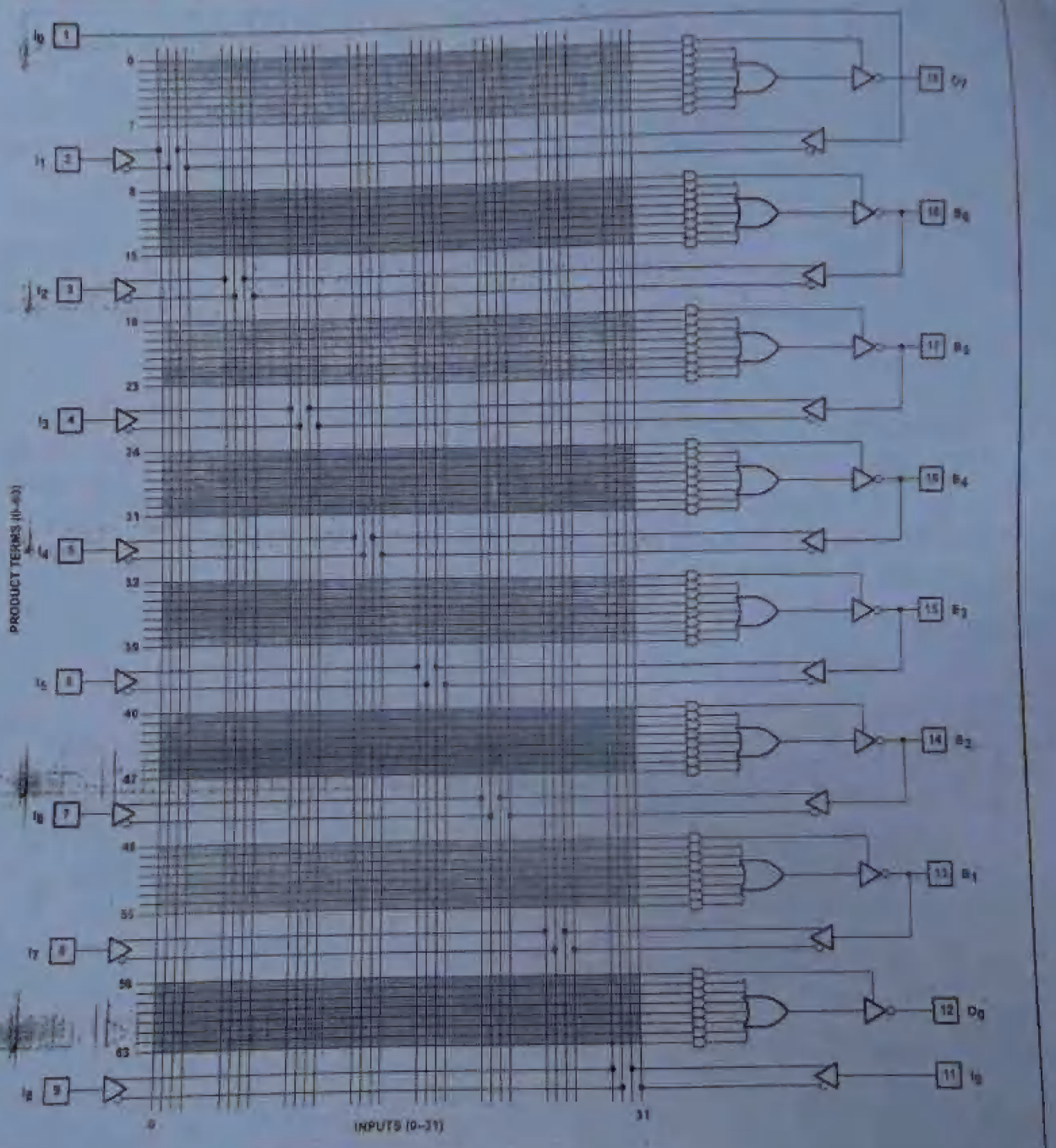
With a "Set/Reset Port C Bit" command, any Port C line programmed as an output (including INTR, IBF and $\overline{OBF}$) can be written, or an interrupt enable flag can be either set or reset. Port C lines programmed as inputs, including $\overline{ACK}$ and $\overline{STB}$ lines, associated with Port C are not affected by a "Set/Reset Port C Bit" command. Writing to the corresponding Port C bit positions of the $\overline{ACK}$ and $\overline{STB}$ lines with the "Set/Reset Port C Bit" command will affect the Group A and Group B interrupt enable flags, as illustrated in Figure 18.

Current Drive Capability

Any output on Port A, B or C can sink or source 2.5 mA. This feature allows the 82C55A to directly drive Darlington type drivers and high-voltage displays that require such sink or source current.

LOGIC DIAGRAM

PLUS16L8



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ⊠ Programmable connections.

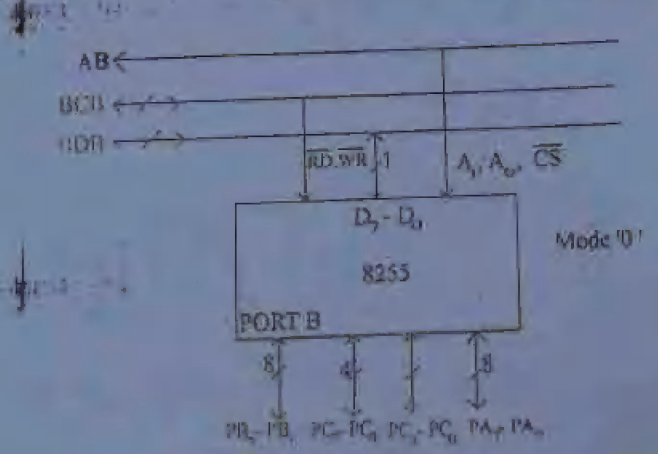
Where are the Handshake signals? We have already listed all the 40 pins of 8255. Port C pins act as handshake signals, when Port A and Port B are configured for other than Mode 0. Port A in Mode 2 and Port B in Mode 1 is possible, as it needs only 5+3 = 8 handshake signals.

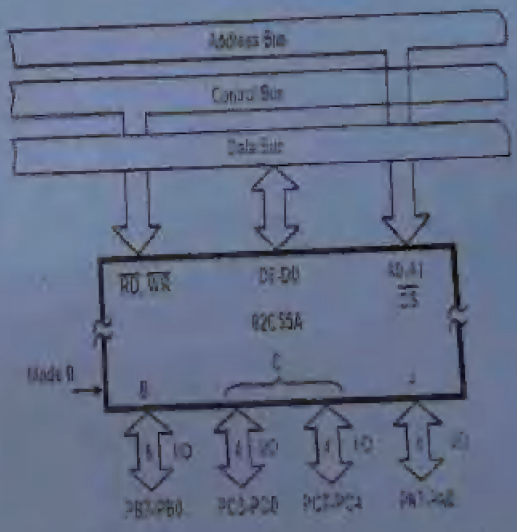
After Reset of 8255, Port A, Port B, and Port C are configured for Mode 0 operation as input ports.

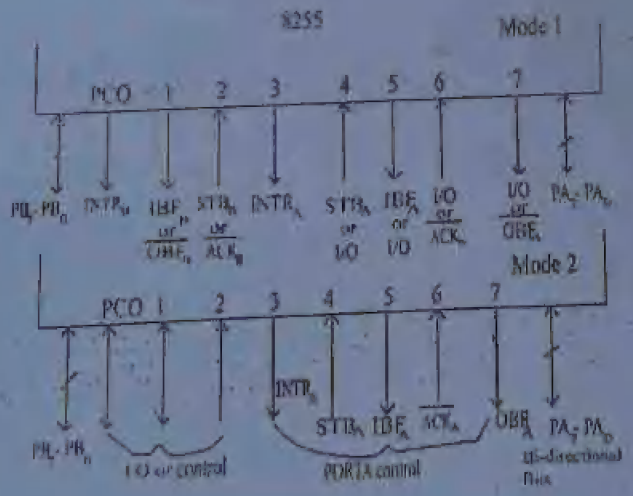
PC2-0 are used as handshake signals by Port B when configured in Mode 1. This is immaterial whether Port B is configured as input or output port.

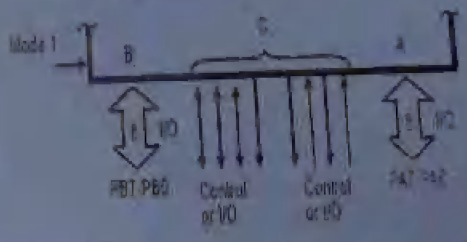
PC5-3 are used as handshake signals by Port A when configured as input port in Mode 1.

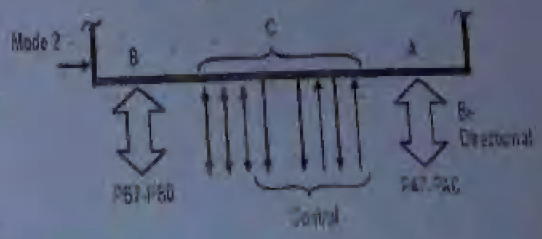
PC7, 6, 3 are used as handshake signals by Port A when configured as output port in Mode 1. PC7-3 are used as handshake signals by Port A when configured in Mode 2.

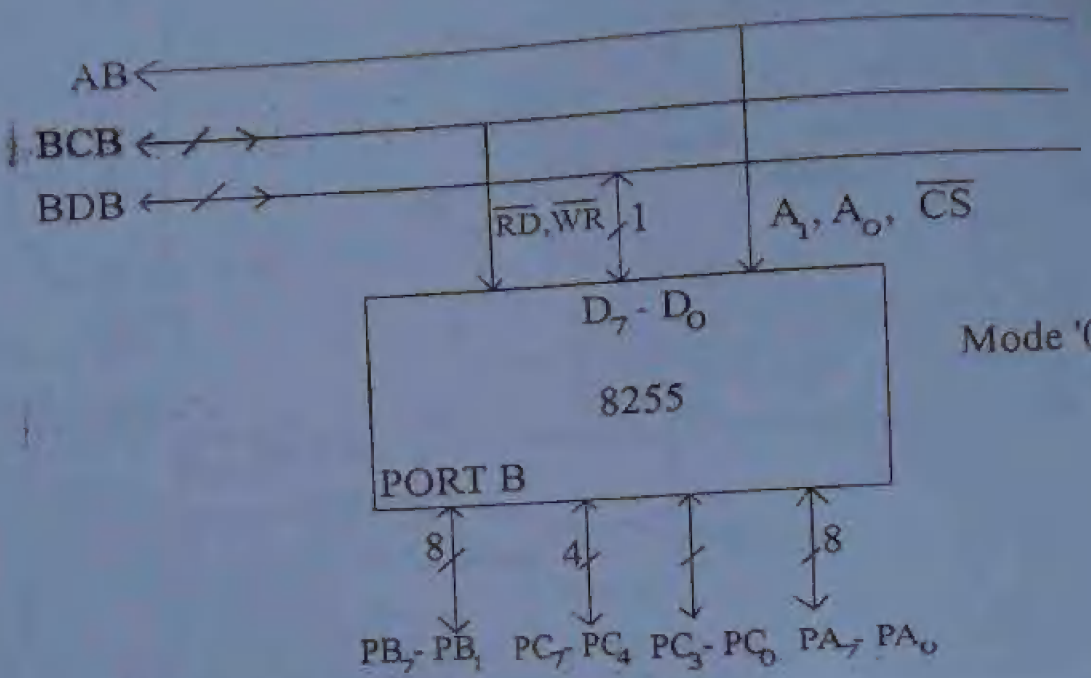


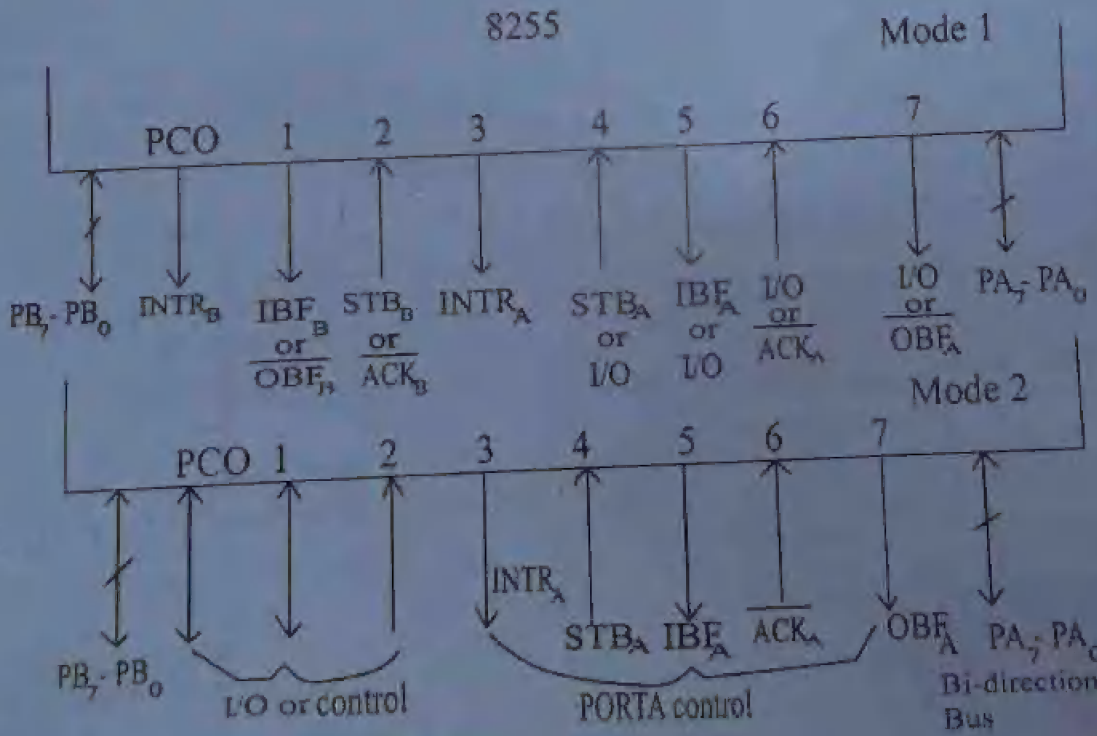




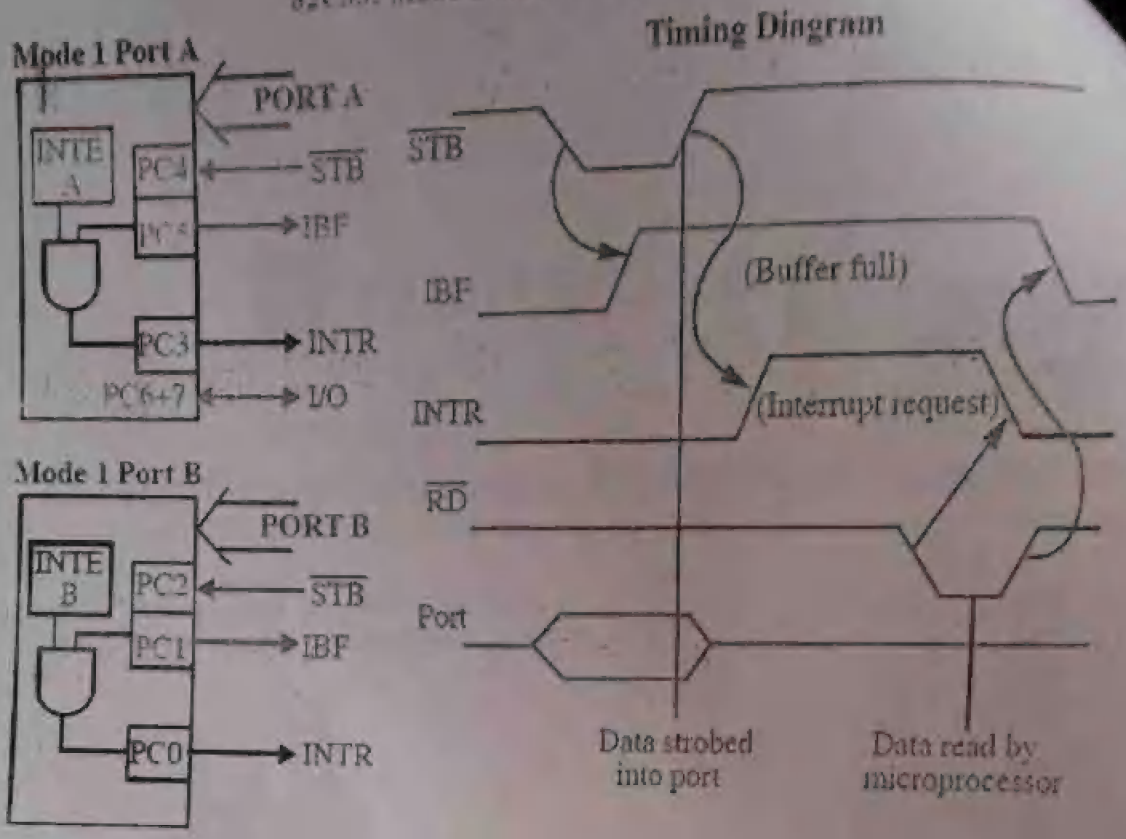




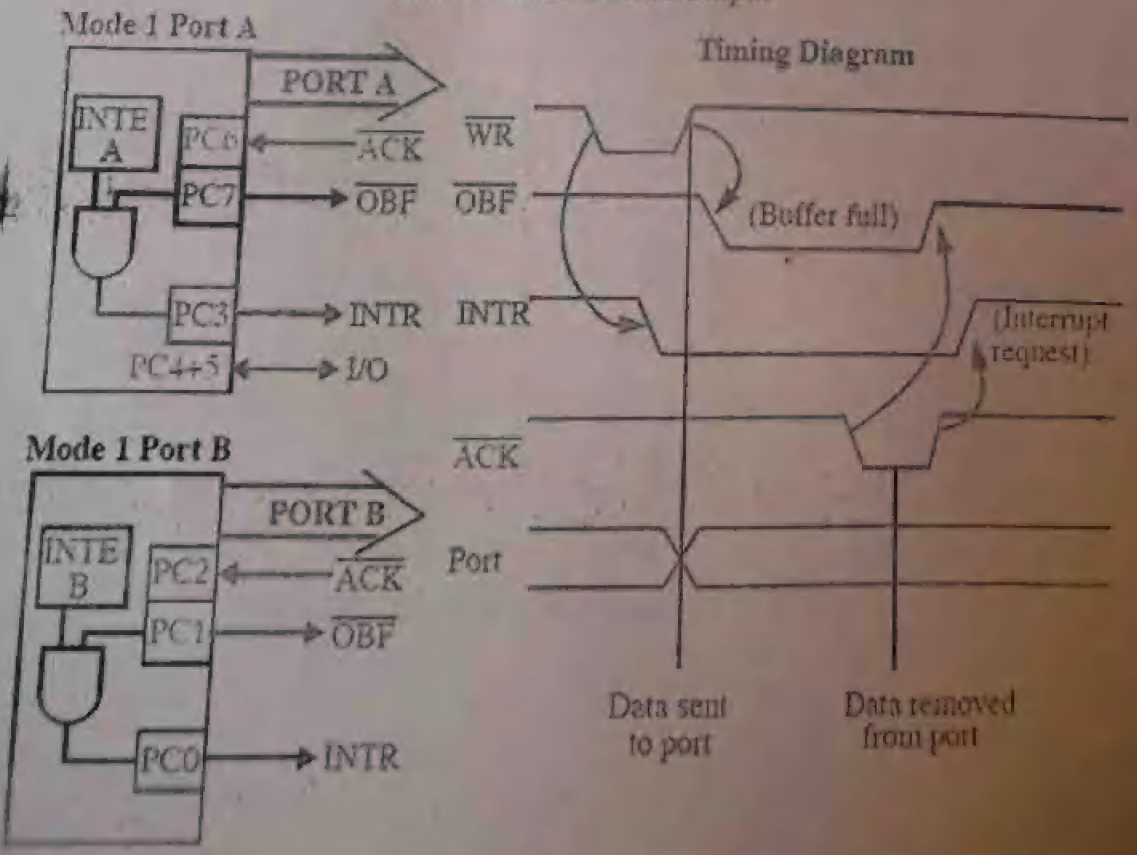

AB
BCB
BDB
RD, WR
1
A1, A0, CS
D7 - D0
8255
PORT B
Mode '0'
8
4
4
8
PB7 - PB1
PC7 - PC4
PC3 - PC0
PA7 - PA0


8255
Mode 1
PCO 1 2 3 4 5 6 7
PB7 - PB0
INTRB
IBFB or OBFB
STBB or ACKB
INTRA
STBA or I/O
IBFA or I/O
I/O or ACKA
I/O or OBFA
PA7 - PA0
Mode 2
PCO 1 2 3 4 5 6 7
PB7 - PB0
I/O or control
INTRA
STBA
IBFA
ACKA
OBFA
PA7 - PA0
PORTA control
Bi-directional Bus

82C55: Mode 1 Strobed Input



82C55: Mode 1 Strobed Output



PERIPHERAL CHIPS

8255: Programmable Peripheral Interface

1. Draw the pin diagram of PPI 8255.

Ans. The pin diagram of 8255 is shown in Fig. 9a.1

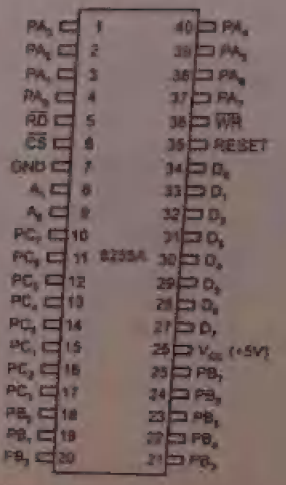


Fig. 9a.1: 8255 Pin diagram
(*Source:* Intel Corporation)

2. Draw the block diagram of 8255.

Ans. The block diagram is shown in Fig. 9a.2.

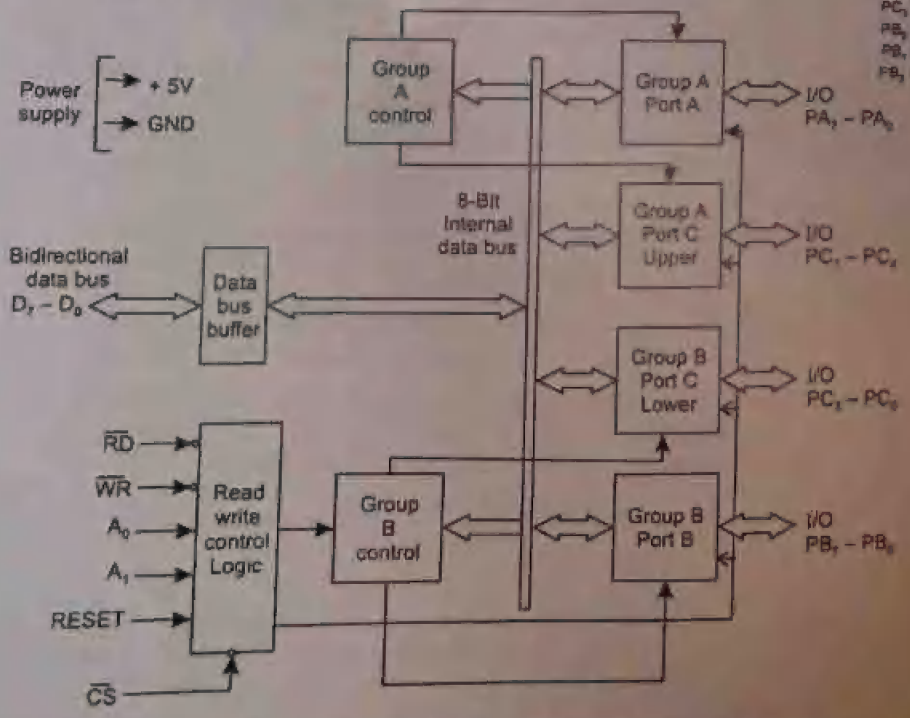


Fig. 9a.2: Block diagram of 8255 (*Source:* Intel Corporation)

3. How many ports are there in 8255 and what are they?

Ans. Basically there are three ports in 8255, viz., Port A, Port B and Port C, each having 8 pins. Again Port C can be divided into Ports C_{upper} and Port C_{lower}—each having four pins i.e., a nibble. Thus 8255 can be viewed to have four ports—Port A, Port B, Port C_{upper} and Port C_{lower}.

4. What pins are associated with Read/Write control logic block?

Ans. There are six pins associated with Read/Write control logic block. These are $\overline{CS}$, $\overline{WR}$, A_0, A_1, Reset and $\overline{CS}$ signals.

5. In how many modes can 8255 operate?

Ans. PPI 8255 can operate in three modes. (*a*) Mode 0 (*b*) Mode 1 and (*c*) Mode 2.
Apart from the above, there is another mode called BSR mode (Bit Set/Reset mode).

6. Distinguish between the three modes of 8255.

Ans. The three modes are Mode 0, Mode 1 and Mode 2. These are I/O operations and selected only if D_7 bit of the control word register is put as 1.

The three operating modes of 8255 are distinguished in the following manner:

Mode 0: This is a basic or simple input/output mode, whose features are:

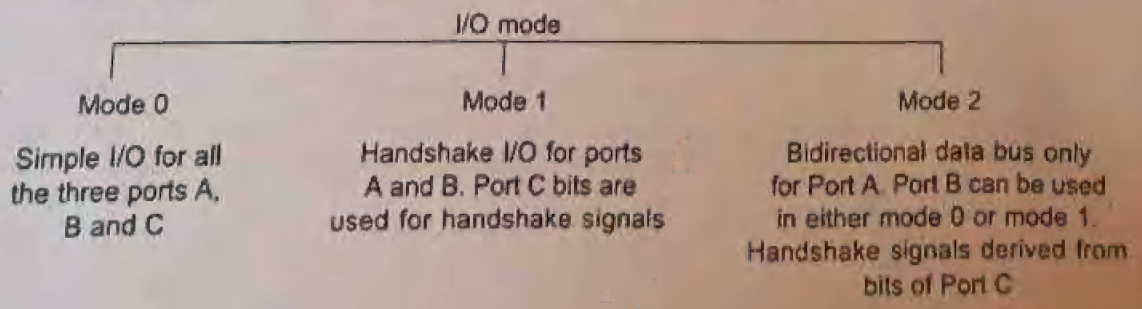
- Outputs are latched.
- Inputs are not latched.
- All ports (A, B, C_U, C_L) can be programmed in either input or output mode.
- Ports don't have handshake or interrupt capability.
- Sixteen possible input/output configurations are possible.

Mode 1: In this mode, input or outputting of data is carried out by taking the help of handshaking signals, also known as strobe signals. The basic features of this mode are:

- Ports A and B can function as 8-bit I/O ports, taking the help of pins of Port C.
- I/Ps and I/Ps are latched.
- Interrupt logic is supported.
- Handshake signals are exchanged between CPU and peripheral prior to data transfer.
- In this mode, Port C is called status port.
- There are two groups in this mode—group A and group B. They can be configured separately. Each group consists of an 8-bit port and a 4-bit port. This 4-bit port is used for handshaking in each group.

Mode 2: In this mode, Port A can be set up for bidirectional data transfer using handshake signals from Port C. Port B can be set up either in mode 0 or mode 1.

The basic operations of the three modes are shown below:



7. Which word determines the operating mode of 8255?

Ans. A single control word determines the operating mode of 8255.

8. For data transfer using 8255, when mode 0 should be selected?

Ans. When unconditional or non-handshaking I/O is required, mode 0 is chosen.

9. How many categories of handshake signals are there? Which is advantageous?

Ans. Handshake signals can be used with either (1) status check I/O or (2) Interrupt I/O.

In the status check I/O, the CPU gets tied up in a loop until the status of the I/O becomes ready while it is the I/O device which interrupts the CPU in interrupt I/O.

10. Explain how the different ports and control words are selected for 8255.

Ans. The two address lines, along with $\overline{CS}$ signal, determine the selection of a particular port or control register. This is explained below:

$\overline{CS}$	A_1	A_0	*Selection*
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Control Register
1	X	X	8255 not selected (because $\overline{CS}$ = 1)

$\overline{CS}$ signal is made 0 by choosing $A_7 = 1$ and A_6 though $A_2 = 0$

Thus,

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0			
1	0	0	0	0	0	0	0	=	80_H	Port A selected
1	0	0	0	0	0	0	1	=	81_H	Port B selected
1	0	0	0	0	0	1	0	=	82_H	Port C selected
1	0	0	0	0	0	1	1	=	83_H	Control Register selected

11. What is BSR mode and what are its characteristics?

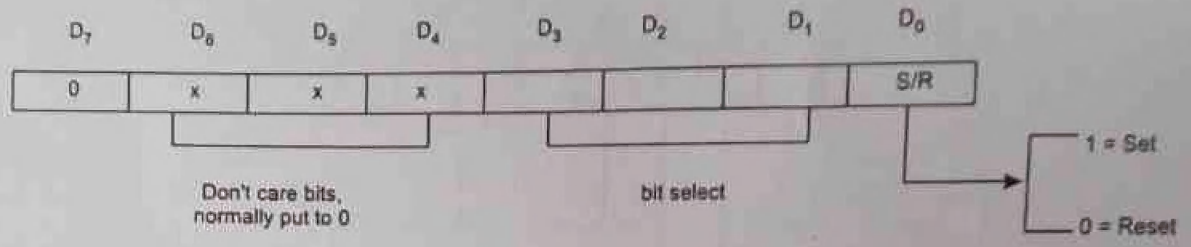
Ans. BSR mode stands for Bit Set Reset mode.

The characteristics of BSR mode are:

- BSR mode is selected only when $D_7 = 0$ of the Control Word Register (CWR).
- Concerned with bits of port C.
- Individual bits of Port C can either be Set or Reset.
- At a time, only a single bit of port C can be Set or Reset.
- Is used for control or on/off switch.
- BSR control word doesn't affect ports A and B functioning.

12. Discuss the control word format in the BSR mode.

Ans. The content of the control word register will be as follows, when used in the BSR mode and selects (either Sets or Resets) a particular bit of Port C at a time.



D_3	D_2	D_1	Particular bit of Port C selected
0	0	0	Bit 0
0	0	1	Bit 1
0	1	0	Bit 2
0	1	1	Bit 3
1	0	0	Bit 4
1	0	1	Bit 5
1	1	0	Bit 6
1	1	1	Bit 7

Fig. 9a.3: The CWR in the BSR mode

13. Write a BSR control word to set bits PC_7 and PC_0 and to reset them after 1 second delay.

Ans. To set or reset any particular bit of Port C in the BSR mode, the control word register is to be appropriately loaded. The above is done by loading the accumulator and sending the same to the control register (i.e., by sending the same to the address of the control register). The address of control word register (CWR) is 83_H.

Program:

MVIA, $0F_H$	(Accumulator loaded with $0F_H$ to set PC_7 bit of Port C)
OUT 83_H	(This sets PC_7 bit of Port C)
MVIA, 01H	(Accumulator loaded with 01_H to set PC_0 bit of Port C)
OUT 83_H	(This sets PC_0 bit of Port C)
CALL DELAY	(Assume the DELAY is for 1 second)
MVIA, 00H	(Accumulator loaded with 00_H to reset PC_0 bit of Port C)
OUT 83_H	(This resets PC_0 bit of Port C)
MVIA, $0E_H$	(Accumulator loaded with $0E_H$ to reset PC_7 bit of Port C)
OUT 83_H	(This resets PC_7 bit of Port C)

14. Show the control word format for I/O mode operation of PPI 8255.

Ans. The control word format, when 8255 is operated in I/O mode, is shown below:

For 8255 PPI to be operated in I/O mode, D_7 bit must be 1.

The three ports are clubbed into two groups—Groups A and B. Group A consists of Port A and C_U. Port A can be operated in any of the modes—0, 1 or 2. Group B consists of Port B and C_L. Here Port B can be operated in either mode 0 or 1.

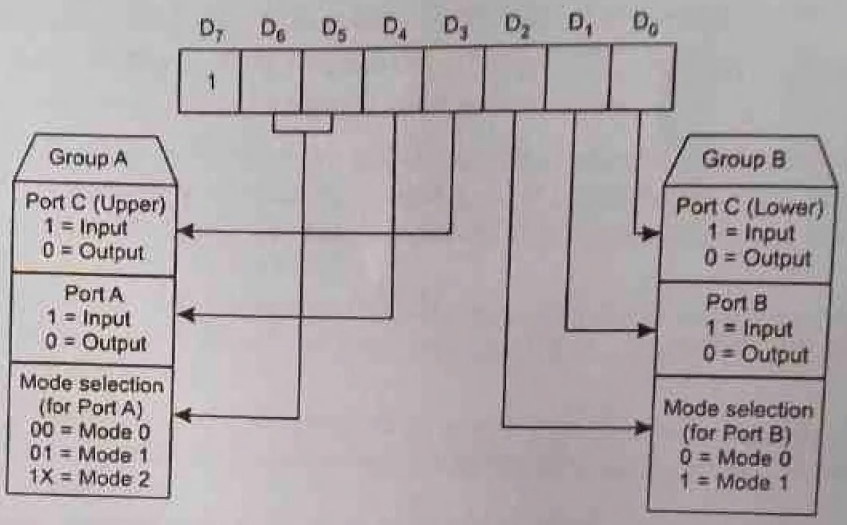


Fig. 9a.4: The CWR in the I/O mode

15. What happens when RESET pin of 8255 is made high?

Ans. When a 1 is applied on RESET pin of 8255, the three ports are put in the input mode. All flip-flops are cleared and interrupts are reset. This condition is not altered even when RESET goes low. 8255 can then be programmed in any mode by appropriately loading the control word register. The mode operation can be changed by altering the content of the control word register, whenever needed.

16. Write down the mode 0 control words for the following two cases:

(a) Port A = Input port, Port B = not used, Port C_U = Input port and Port C_L = Output port.

(b) Port A = Output port, Port B = Input port, Port C = Output port

Ans. The control words for the two cases will be as follows:

(a)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
1	0	0	1	1	0	0	0	=	98_H
I/O mode	Mode 0 for Port A		Port A input	Port C_U input	Port B not used		Port C_L output		

Thus, the control word would be = 98_H

(b)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
1	0	0	0	0	0	1	0	=	82_H
I/O mode	Mode 0 for Port A		Port A output	Port C_U output	Mode 0 for Port B	Port B input	Port C_L output		

Thus, the control word would be = 82_H.

17. In mode 1 what are the control signals when ports A and B act as input ports. Discuss the control signals. Draw the timing waveforms for such a strobed input.

Ans. The following are the control signals when ports A and B act as input ports (under mode 1) $\overline{STB_A}$, IBF_A, $INTE_A$ for Port A and $\overline{STB_B}$, IBF_B, $INTE_B$ for Port B, respectively. The details about the input control signals are discussed below:

- $\overline{STB}$ *(Strobe input):* This is an active low signal generated by a peripheral device. When a peripheral device has some valid data, it sends the same via Port A or B and sends a low $\overline{STB}$ signal. This data is accepted by 8255 and it generates a IBF and INTR (provided INTE is set previously).
- *IBF (Input buffer full):* On receipt of $\overline{STB}$ signal from peripheral device, data is stored in 8255 by its input latch. In its turn, 8255 generates a high IBF. IBF is reset when CPU reads the data.
- *INTR (Interrupt request):* This active high output signal is generated only if $\overline{STB}$, IBF and INTE are all set at the same time. This signal interrupts the CPU via its INTR (pin no. 10 of 8085).
- *INTE (Interrupt Enable):* This is an internal F/F which can be set/reset using the BSR mode. It must be set if INTR signal is to be effective.

The following figure shows Port A in Mode 1 (input), along with the timing diagrams.

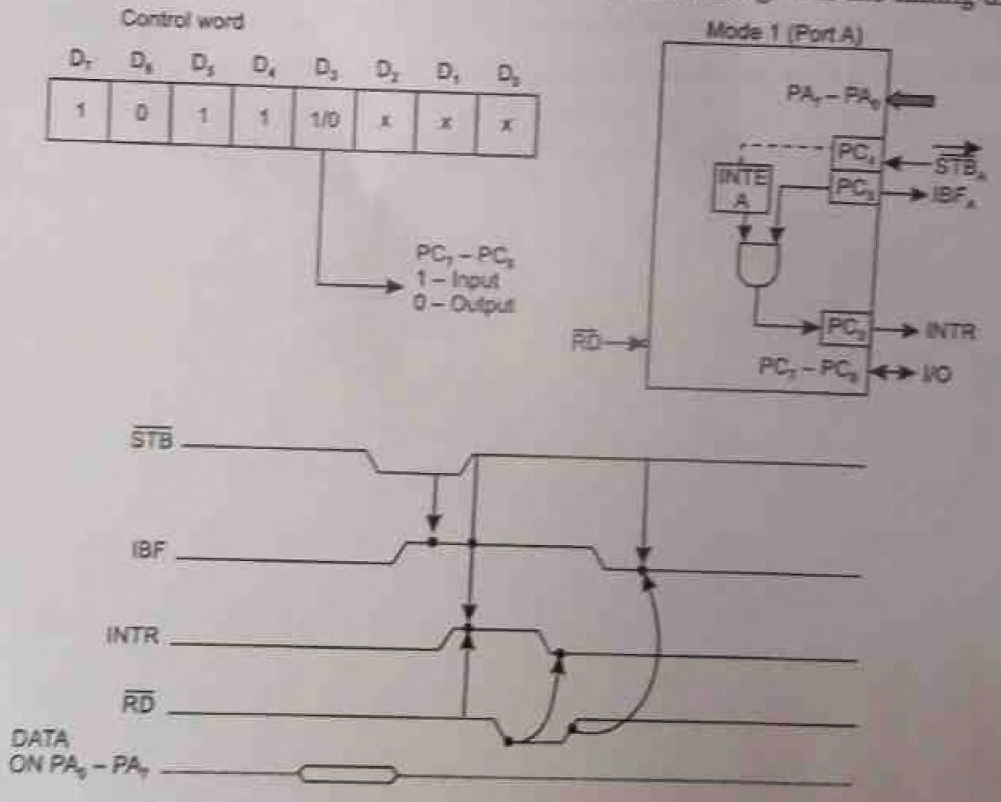


Fig. 9a.5: Port A in Mode 1 (Input) (*Source:* Intel Corporation)

18. *In mode 1, what are the control signals when ports A and B act as output ports. Discuss the control signals. Draw the timing waveforms for such a strobed output.*

Ans. The following are the control signals when ports A and B act as output ports (under mode 1) $\overline{OBF_A}$, $\overline{AC}_A$, $INTE_A$ for Port A and $\overline{OBF_B}$, $\overline{ACK_B}$, $INTE_B$ for Port B respectively.

The details about the output control signals are discussed below:

- $\overline{OBF}$ *(Output buffer full):* This is an active low output signal. This signal becomes low when the CPU writes data into the output latch of 8255. This output signal from 8255, which goes to a peripheral, indicates to the peripheral that the data on the output latch of 8255 is ready to be read.
- $\overline{ACK}$ *(Acknowledge):* When data reading by the peripheral from the output latch of 8255 is complete (i.e., the peripheral has accepted the data), it (the peripheral) outputs a low signal which is connected to the $\overline{ACK}$ (input signal) pin of 8255. On receipt of this low signal by 8255 (from peripheral), the $\overline{OBF}$ line of 8255 goes high.
- *INTR (Interrupt):* This signal is set only if $\overline{OBF}$, $\overline{ACK}$ and INTE (internal F/F) are all at high(1) state. This output signal from 8255 goes to INTR (pin 10 of 8085) to interrupt the CPU. The INTR signal is reset on the falling edge of $\overline{WR}$.
- *INTE (Interrupt Enable):* This is an internal F/F which can be set/reset in BSR mode. This must be set if INTR signal is to be effective.

The following figure shows Port A in mode 1 (output), along with the timing waveforms.

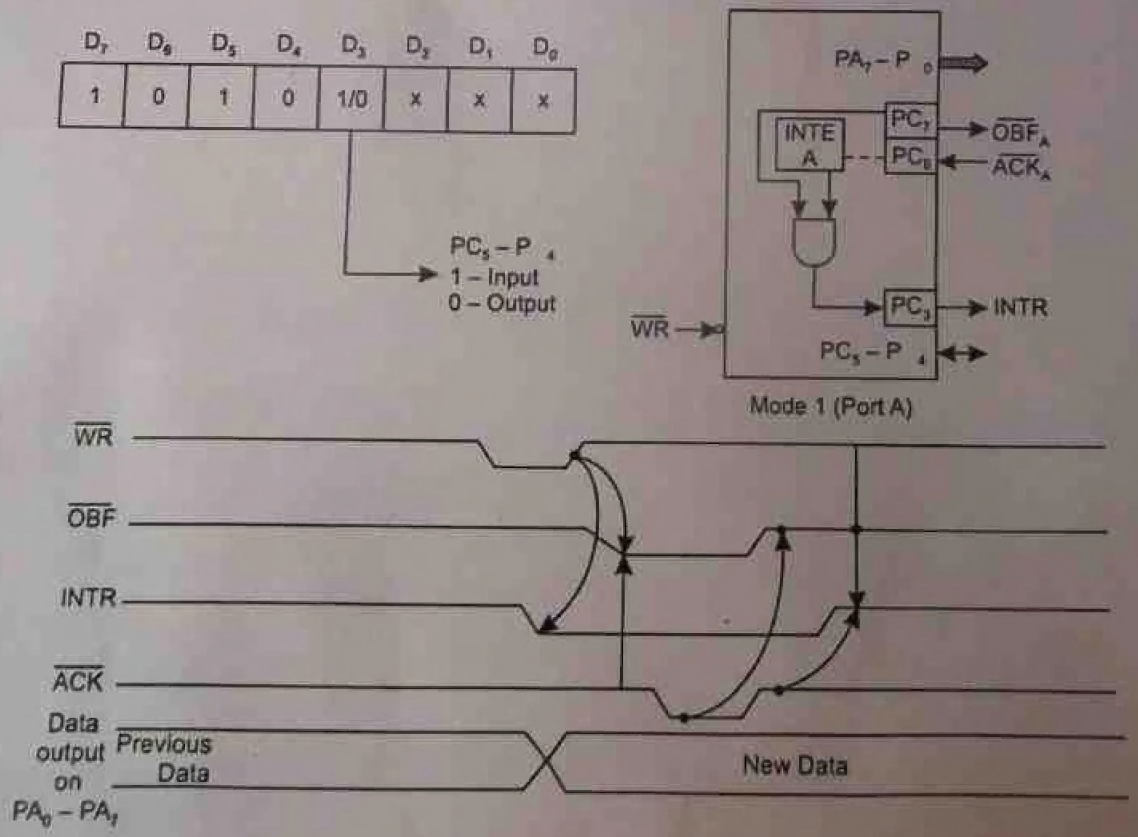


Fig. 9a.6: Port A in mode 1 (Output) (*Source*: Intel Corporation)